Intelligent Systems with Applications 19 (2023) 200253

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

Review

Meta-heuristic approaches for the University Course Timetabling Problem

Sina Abdipoor [a,*], Razali Yaakob [a], Say Leng Goh [b,*], Salwani Abdullah [c]

[a] Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
[b] Optimisation and Visual Analytics Research Group, Faculty of Computing and Informatics, Universiti Malaysia Sabah Labuan International Campus, 87000 Labuan, Malaysia
[c] Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia

ARTICLE INFO

Keywords:
Operational research
Combinatorial optimization
University Course Timetabling Problem
Meta-heuristics
Hybrid meta-heuristics

ABSTRACT

Course timetabling is an ongoing challenge that universities face all around the world. This combinatorial optimization task involves allocating a set of events into finite time slots and rooms while attempting to satisfy a set of predefined constraints. Given the high number of constraints and the large solution space to be explored, the University Course Timetabling Problem (UCTP) is classified as an NP-hard problem. Meta-heuristic approaches have been commonly applied to this problem in the literature and have achieved high performance on benchmark datasets. This survey paper provides a comprehensive and systematic review of these approaches in the UCTP. It reviews, summarizes, and categorizes the approaches, and introduces a classification for hybrid meta-heuristic methods. Furthermore, it critically analyzes the benefits and limitations of the methods. It also presents challenges, gaps, and possible future work.

1. Introduction

The Educational Timetabling Problem (ETP) is an open-ended, demanding administrative task that frequently occurs in most academic institutions (Tan et al., 2021, Thepphakorn & Pongcharoen, 2019, Silva et al., 2021). The objective of this Combinatorial Optimization Problem (COP) (Blum et al., 2011, Sabar et al., 2021, Ngoo et al., 2022) is to assign resources in time and space in such a way that satisfies stakeholders' requirements and increases utilization (Lindahl et al., 2018, Goh et al., 2020, Abdelhalim & El Khayat, 2016). Educational timetabling can be classified into university and (high) school timetabling. University timetabling is further divided into the University Course Timetabling Problem (UCTP or UCTTP) and the University Examination Timetabling Problem (UETP or UETTP) (Goh et al., 2019b, Rezaeipanah et al., 2021, Akkan & Gülcü, 2018). Fig. 1 illustrates the problem diagram for the UCTP.

The UCTP has drawn great interest from researchers of various fields (Teoh et al., 2015). This task needs to be repeatedly performed at the beginning of each academic year (semester) at universities (Tan et al., 2021, Thepphakorn & Pongcharoen, 2019, Abdelhalim & El Khayat, 2016, Rezaeipanah et al., 2021). Given a set of events (lectures, students, and professors), finite resources (rooms and facilities), and time slots (time periods across the weekdays), the UCTP can be defined as the assignment of E events to R rooms and T time slots in compliance with a set of optional and mandatory constraints (Goh et al., 2020, Lewis & Thompson, 2015). This problem is a special case of the Graph Coloring Problem in which events and time slots are represented by vertices and edges, respectively (Lewis & Thompson, 2015). As there exist R^E ways of allocation in the UCTP (Tindell et al., 1992), the computational time increases exponentially with the growth in problem size. Thus, the UCTP is regarded as a Non-deterministic Polynomial-time hard (NP-hard) problem (Chen et al., 2021, Babaei et al., 2015, Song et al., 2018, NoorianTalouki et al., 2022, Hosseini Shirvani & Noorian Talouki, 2022). This makes the application of exact algorithms infeasible, especially on larger problems (Schaerf, 1999).

Another challenge in the UCTP is the development of an approach with high general applicability, capable of being easily applied to different instances and problems (Blum et al., 2011, Goh et al., 2019b, Rezaeipanah et al., 2021, Akkan & Gülcü, 2018, Bashab et al., 2020, Shirvani & Talouki, 2021). The lack of general applicability in the literature necessitates manual timetabling (Chen et al., 2021), which is extremely difficult, time-consuming, and often leads to the wastage of resources (Thepphakorn & Pongcharoen, 2019, Abdelhalim & El Khayat, 2016).

* Corresponding authors.
E-mail addresses: abdipoor.sina@student.upm.edu.my (S. Abdipoor), razaliy@upm.edu.my (R. Yaakob), slgoh@ums.edu.my (S.L. Goh), salwani@ukm.edu.my (S. Abdullah).

https://doi.org/10.1016/j.iswa.2023.200253
Received 1 February 2022; Received in revised form 25 May 2023; Accepted 17 June 2023
Available online 22 June 2023

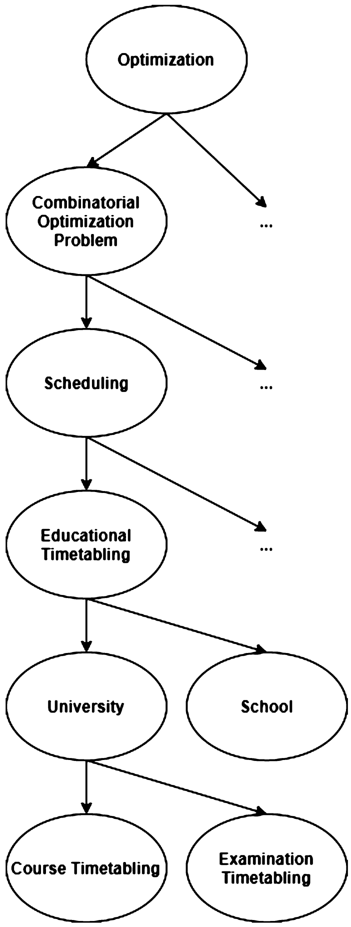

Fig. 1. UCTP problem diagram.

Meta-heuristic approaches have emerged as effective solutions to address these challenges, as they excel in searching large solution spaces and handling diverse problem instances (Blum et al., 2011, Teoh et al., 2015, Ilyas & Iqbal, 2015, Goh et al., 2022). These methods are widely employed in the literature and have demonstrated high performance on benchmark datasets for the UCTP (Silva et al., 2021, Chen et al., 2021, Babaei et al., 2015, Bashab et al., 2020, Bettinelli et al., 2015).

Numerous survey papers on the UCTP have been published to date. Table 1 provides a chronological summary of these papers since 2015. However, most of these papers primarily offer a general overview of the methodologies applied to the UCTP, often overlooking critical analysis and discussion of these methods. Additionally, a comprehensive review, comparison, and classification of (hybrid) meta-heuristics are lacking.

This paper aims to fill this gap by focusing on meta-heuristic and hybrid meta-heuristic approaches for the UCTP. These approaches are thoroughly reviewed, and their methods are classified, analyzed, and compared.

The main contributions of this paper are:

1. Presenting a thorough overview of the UCTP and its benchmark datasets;
2. Classifying the literature based on the problem variant (CB-CTP/PE-CTP);
3. Introducing a categorization for hybrid meta-heuristic approaches based on their type of hybridization (collaborative and integrative);
4. Reviewing and critically analyzing meta-heuristic and hybrid meta-heuristic approaches in the literature;
5. Identifying trends, strengths, and limitations of the approaches;
6. And suggesting future research directions based on the findings.

The rest of this paper is organized as follows. Section 2 presents the methodology used for this survey paper. Section 3 gives a comprehensive overview of the UCTP, its variants, and the benchmark datasets. Section 4 provides a detailed review and analysis of meta-heuristic approaches. Section 5 reviews and categorizes the hybrid meta-heuristic approaches in the literature. Strengths, limitations, and trends of different meta-heuristic and hybrid meta-heuristic approaches are presented in the discussions of Sections 4 and 5, respectively. Possible future research opportunities in the UCTP are suggested in Section 6. Finally, Section 7 concludes the findings.

2. Survey methodology

This survey paper conducts a systematic literature review of meta-heuristic and hybrid meta-heuristic approaches. Firstly, a large-scale search of the literature was conducted on online databases using different combinations of relevant keywords to retrieve all relevant papers published from 2015 onwards. This time frame was chosen as several review papers were published in that year, as indicated in Table 1. Moreover, the popularity of meta-heuristic and hybrid meta-heuristic approaches is evident, as emphasized in Chen et al. (2021), further underscoring the need for a detailed study of these approaches. The search strategy used for this survey paper is presented in Table 2. Secondly, a process of inclusion and exclusion was applied to filter these papers in different stages. Finally, the methodologies of the selected papers were categorized and summarized in tables and figures.

All collected papers undergo four filtering stages to identify the most appropriate bibliography. Table 3 presents the number of papers after each of these filtering stages. In stage 1, all the retrieved papers are filtered based on their title and authors, and duplicate items are removed. Table 4 classifies the papers at this stage based on their publication year. It can be seen that the UCTP is still a highly active research field. In stage 2, the abstracts are reviewed, and the papers are filtered based on their problem. Fig. 2 categorizes the papers based on their problem type. Papers irrelevant to the UCTP are discarded at this stage. In stage 3, all the remaining papers are collected, studied, labeled, and filtered based on their methodology. As the scope of this survey paper is meta-heuristics, papers addressing other approaches are cast aside at this stage. Fig. 3 depicts the summary of approaches utilized to tackle the UCTP at stage 2. It can be seen that among all the meta-heuristics, Evolutionary Algorithms (EA), Swarm Intelligence (SI), and hybrid methods are the frequent methodologies in the literature. Other approaches include hyper-heuristics and mathematical approaches. Finally, in stage 4, all remaining papers undergo detailed analysis. Information such as research gap, methodology, dataset, measurement, performance, limitations, and research opportunities are extracted from various sections of the papers. Through citation backtracking, all relevant and missing papers are added to our library. Papers with low comprehensiveness or competitiveness are excluded, with higher emphasis given to papers published in more established journals. Table 5 shows the list of journals of the final selected papers.

3. University course timetabling

3.1. Problem definition

University course timetabling varies considerably in different countries and institutions (Lindahl et al., 2018). This can be attributed to the unique problems each university faces. Thus, various requirements and policies are set by different institutions and the country's education system (Chen et al., 2021).

Many different algorithms have been developed over the years for different variants of this problem (Gülcü & Akkan, 2020). This makes it extremely difficult for researchers to compare their works and assess the performance of their methodology (Lindahl et al., 2018). To address this issue, much effort has been made, and through the introduction of

Table 1
Summary of UCTP survey papers.

Year	Authors	Title	Scope	Limitation
2015	Babaei et al. (2015)	A survey of approaches for university course timetabling problem	A survey of all approaches in the UCTP, focusing on distributed multi-agent systems approaches.	Detailed performance comparison of approaches on benchmarks is missing.
2015	Bettinelli et al. (2015)	An overview of curriculum-based course timetabling	A detailed analytical review of approaches in the CB-CTP.	Only focuses on one variant of the UCTP (CB-CTP), and lacks the review of many meta-heuristics applied to other variants.
2015	Ilyas and Iqbal (2015)	Study of hybrid approaches used for university course timetable problem (UCTP)	Classifying hybrid approaches into local search or population-based + local search-based approaches.	Recent state-of-the-art hybrid meta-heuristics and problem classification are missing.
2015	Teoh et al. (2015)	Review of state of the art for metaheuristic techniques in Academic Scheduling Problems	Studying the properties and complexity of academic scheduling problems and reviewing solution optimality of meta-heuristic approaches.	Lacks methods applied to benchmark datasets or reports their solution quality. It also does not cover some recent meta-heuristic methods.
2016	Pandey and Sharma (2016)	Survey on university timetabling problem	A detailed introduction on the UCTP and a brief review of all approaches.	Analysis of methods and review of hybrid approaches are overlooked
2019	Oude Vrielink et al. (2019)	Practices in timetabling in higher education institutions: a systematic review	A systematic literature review aiming to identify similarities and differences in the theory and practice of timetabling in higher education.	Does not cover the hybrid meta-heuristic approaches that have been proposed to solve the UCTP.
2020	Bashab et al. (2020)	A systematic mapping study on solving university timetabling problems using meta-heuristic algorithms	A mapping study to show the intensity of meta-heuristic publications in the UCTP.	Lacks critical analysis and detailed method classification.
2021	Chen et al. (2021)	A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities	Providing a general overview of all approaches in the UCTP and identifying trend and gaps.	Limited reviewed hybrid approaches without classifying hybrid meta-heuristics in the literature.

Table 2
Search strategy.

Keywords	University Course Timetabling, Hybrid, Meta-heuristic
Year	2015-2022
Online Tools/Databases	Google Scholar, Elsevier, Springer, IEEE

Table 3
Papers filtering.

Stage Number	Number of Papers
0	151
1	134
2	108
3	77
4	45

Table 4
Papers publication year at stage 1.

Publication Year	Number of Papers
<2015	9
2015	17
2016	15
2017	15
2018	16
2019	13
2020	18
2021	17
2022	14
Total	**134**

Table 5
List of journals of the final selected papers.

Journal	Paper Count
European Journal of Operational Research (EJOR)	3
Computers & Operations Research (COR)	3
Applied Soft Computing	2
Expert Systems with Applications	2
Others	35

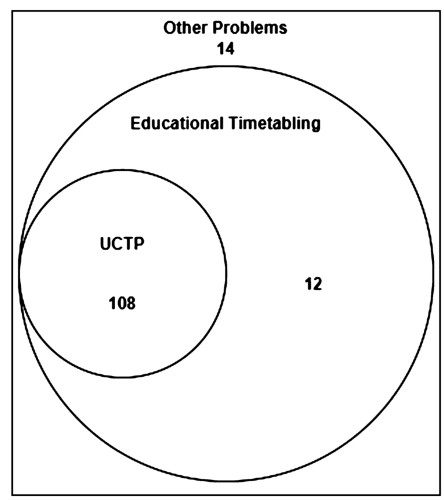

Fig. 2. Papers problem category at stage 1.

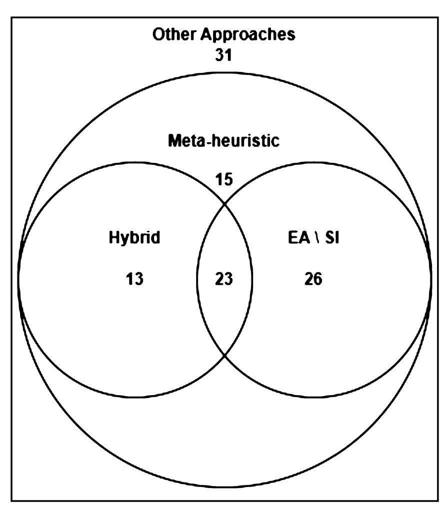

Fig. 3. Papers approach at stage 2.

the International Timetabling Competition (ITC), the standard characterization of UCTP was presented.

The standard UCTP is based on a conventional definition of timetabling from 1995 (Wren, 1995). It is a simplified and abstracted model of the real-world problem that aims to capture its essential features (Rezaeipanah et al., 2021, Akkan & Gülcü, 2018). The standard UCTP can be formally stated as follows (Goh et al., 2019b):

Given: a set of events, $E = \{e_1, e_2, e_3, ..., e_{|E|}\}$
a set of time slots, $T = \{t_1, t_2, t_3, ..., t_{|T|}\}$ ($|T| = 45$ in benchmark datasets - 9 time slots per day $*$ 5 days per week)
a set of rooms $R = \{r_1, r_2, r_3, ..., r_{|R|}\}$
a set of students $S = \{s_1, s_2, s_3, ..., s_{|S|}\}$
a set of features $F = \{f_1, f_2, f_3, ..., f_{|F|}\}$
and a set of days $D = \{d_1, d_2, d_3, ..., d_{|D|}\}$ (D is commonly considered as the weekdays, $|D| = 5$)

Find: an assignment (a timetable) of E events (with S students) to R rooms (with F features) and T time slots (across D days) that minimizes constraint violations

The formal mathematical formulation of UCTP constraints is presented in Lindahl et al. (2018), Goh et al. (2019b), Teoh et al. (2015), Lewis and Thompson (2015), Pandey and Sharma (2016).

Constraints in UCTP are generally classified into Hard Constraints (HC) and Soft Constraints (SC) (Thepphakorn & Pongcharoen, 2019, Goh et al., 2020, 2019b, Rezaeipanah et al., 2021, Babaei et al., 2015, Song et al., 2018). While hard constraints are compulsory restrictions that determine the feasibility of a given solution, soft constraints are optional and ascertain the quality of a solution (Chen et al., 2021, Goh et al., 2019a). In many scenarios (theoretical and real-world), a solution that violates any of the hard constraints (an infeasible solution) is considered worthless (Chen et al., 2021).

3.2. Problem variants

The unique necessities and requirements of various universities imply different constraints and objectives, leading to distinct variants of the university timetabling problem. The UCTP is commonly divided into two subcategories, the Curriculum-Based Course Timetabling Problem (CB-CTP) and the Post-Enrollment Course Timetabling Problem (PE-CTP) (Akkan & Gülcü, 2018, Teoh et al., 2015, Chen et al., 2021, Bettinelli et al., 2015). The major difference between these two is their source of conflict, i.e., conflicts in the CB-CTP arise from the published curriculum, while in the PE-CTP, they primarily originate from students' enrollment data (Song et al., 2021). Each university might opt for one of the variants based on its organization. However, both CB-CTP and PE-CTP are significant in real-world applications (Bettinelli et al., 2015).

CB-CTP and PE-CTP were formally defined and distinguished in two different tracks in the International Timetabling Competition 2007 (Di Gaspero et al., 2007). In the CB-CTP, a course consists of a set of lectures, which are predefined in a curriculum. In the PE-CTP, however, each course is a single event (Bettinelli et al., 2015). In the CB-CTP, the curricula of the students are known, but not the student enrollment. Meanwhile, students' enrollment occurs prior to the timetabling process in the PE-CTP (Soria-Alcaraz et al., 2016). Although it is shown that these variants are closely related (Lewis & Thompson, 2015), a distinctive feature between these variants is that only the PE-CTP involves student sectioning, i.e., the possibility of "assigning students to individual sections of a course" (Müller & Murray, 2010). Consideration of student sectioning is essential but increases complexity (Bettinelli et al., 2015).

Depending on the unique demands arising in real-world and theoretical applications, distinct variants of the UCTP with a different set of requirements and constraints exist. Much research has addressed these alternative variants. Babaei et al. addressed the problem of common lectures among different departments in a real-world application (Babaei et al., 2019). In related research (Song et al., 2017), energy efficiency was incorporated as an objective on the dataset collected from the Liberal Arts Building 1 at Seoul National University. Aschinger et al. introduced several new constraints and features into the International Timetabling Competition 2007 dataset to cope with the real-world terms at University College London (UCL) (Aschinger et al., 2018). Related research (Thepphakorn et al., 2020, 2021) attempted to minimize the total operating cost. Kasemset et al. included a predefined pattern of days and time slots in the standard UCTP (Kasemset & Irohara, 2019). Fairness was used as an objective to address a real-world UCTP from Caraga State University. Gozali et al. focused on the student sectioning problem (Gozali et al., 2020). And robustness was introduced as a new measurement to address the UCTP in Akkan and Gülcü (2018) and Gülcü and Akkan (2020). A comprehensive systematic study on different subproblems of UCTP can be found in Herres and Schmitz (2021).

3.3. Constraints

The constraints involved in the UCTP include:

- Hard Constraints (HC):
  - **HC1:** No student can be assigned more than one course at the same time.
  - **HC2:** The room should satisfy the features required by the course.
  - **HC3:** The number of students attending the course should be less than or equal to the capacity of the room.
  - **HC4:** No more than one course is allowed for each time slot in each room.
  - **HC5:** A course can only be assigned to some preset time slots.
  - **HC6:** Where specified, a course should be scheduled to occur in the correct order.
  - **HC7:** All lectures of a course must be scheduled. A violation occurs if a lecture is not scheduled.
  - **HC8:** Lectures of courses in the same curriculum or taught by the same teacher must be all scheduled in different periods.
  - **HC9:** If the teacher of the course is not available to teach that course at a given period, then no lectures of the course can be scheduled at that period.
- Soft Constraints (SC):
  - **SC1:** A student should not have a single course on a day.
  - **SC2:** A student should not have more than two consecutive courses.
  - **SC3:** A student should not have a course scheduled in the last time slot of the day.
  - **SC4:** The number of students attending the course should be less than or equal to the capacity of the room (same as HC3 but is considered a soft constraint in the ITC2007-Track3).
  - **SC5:** The lectures of each course must be spread into the given minimum number of days.
  - **SC6:** Lectures belonging to a curriculum should be adjacent to each other (i.e., in consecutive periods).
  - **SC7:** All lectures of a course should be given in the same room.

3.4. Datasets

Different implementations tackling distinct variants of the UCTP have reported varying degrees of success. However, the effectiveness comparison of different algorithms is difficult if they are executed on different problem instances. Standard datasets enable fair comparison and assessment of different algorithms. Datasets used in the literature can be divided into benchmark and real-world datasets (Chen et al., 2021).

3.4.1. Benchmark

Benchmark datasets aim to unify the research in the UCTP by proposing a consolidated formulation of the problem and suggest-

Table 6
ITC winners.

	Rank 1	Rank 2	Rank 3
ITC2002	SA (Meta-heuristic) (Kostuch, 2003)	TS (Meta-heuristic) (Cordeau et al., 2003)	GD (Meta-heuristic) (Bykov, 2003)
ITC2007-Track2	LS-based (Hybrid Meta-heuristic) (Cambazard et al., 2007)	TS-based (Hybrid Meta-heuristic) (Atsuta et al., 2008)	LS-based (Hybrid Meta-heuristic) (Chiarandini et al., 2008)
ITC2007-Track3	GD-based (Hybrid Meta-heuristic) (Müller, 2009)	TS-based (Meta-heuristic) (Lü & Hao, 2010)	TS-based (Hybrid Meta-heuristic) (Atsuta et al., 2008)
ITC2019	MIP (Mathematical) (Holm et al., 2019)	MIP (Mathematical)	SA-based (Meta-heuristic) (Gashi & Sylejmani, 2019)

ing a standard dataset for benchmarking and comparing different approaches. Common benchmark datasets in the literature include:

- International Timetabling Competition: PATAT (Practice And Theory of Automated Timetabling) is a conference series addressing timetabling problems. This conference, which is held every two years, plays a vital role in motivating research in the field. Through the organization of the International Timetabling Competition (ITC) by PATAT and the Metaheuristic Network, standard experimentation and problem formulation of the UCTP were established in 2002 (Pandey & Sharma, 2016). As evident in Table 10, ITC datasets are the most commonly used benchmark datasets in the literature.

  ITC2002 dataset is the first ITC held in 2002. It has 20 instances that were generated by Ben Paechter. To adhere to fair comparison, a time limit is benchmarked for a given machine by running a program on a host computer. This benchmark dataset is available on the ITC2002 website.[1] The hybrid simulated annealing-based approach proposed in Goh et al. (2020) appears to be one of the best-performing approaches post-competition.

  ITC2007 dataset is a further development of educational timetabling. The time limit is benchmarked in the same way as ITC2002. ITC2007 distinguished the different variants of the educational timetabling problems and introduced three distinct datasets for UETP, PE-CTP, and CB-CTP, respectively. Track 2 (PE-CTP) includes 24 instances. Meanwhile, track 3 focuses on the CB-CTP problem variant (Bettinelli et al., 2015) and consists of 21 instances. The benchmark datasets for all three tracks can be downloaded from the ITC2007 website.[2] The current state-of-the-art methods on this dataset include Nagata (2018) and Goh et al. (2020) for track 2, and Kampke et al. (2019) for track 3.

  ITC2019 dataset is the latest ITC competition by PATAT, co-organized by UniTime[3] (an open-source, comprehensive educational scheduling system that supports developing course and exam timetables). Student sectioning combined with standard time and room assignment of events in courses is the key novelty of this dataset, which makes it more complex than the previous datasets. Benchmarking and solution validation, alongside the 30 instances, are available on the ITC2019 website.[4] More information about this dataset can be found in Müller et al. (2018). The low number of publications (see Table 10), coupled with the higher complexity of this dataset, has created a gap for future research to focus on. Table 6 summarizes the winners of ITC2002, ITC2007, and ITC2019. It can be seen that meta-heuristic and mathematical approaches have achieved the highest performance in these competitions.
- Socha: The 11 instances (5 small, 5 medium, and 1 large) of this benchmark dataset were generated by an algorithm developed by Ben Paechter. Unlike the ITC datasets, the time limits in Socha are statically set to 90, 900, and 9000 seconds for the small, medium, and large instances, respectively. Further information and the dataset are available on the website.[5] The proposed methods in Goh et al. (2020) and Nagata (2018) are the current post-competition state-of-the-art.
- Hard: The Hard dataset was created by Lewis and Paechter (2007) and includes 60 instances (20 small, 20 medium, and 20 large). The time limits are set to 30, 200, and 800 seconds for the small, medium, and large instances, respectively. This benchmark dataset[6] focuses on hard constraints and finding feasible solutions (a feasible solution is one that satisfies all the hard constraints (Goh et al., 2019a)). Approaches in Song et al. (2018) and Goh et al. (2020) have managed to outperform other methods on this dataset and appear to be among the best-performing.

Table 7 summarizes the features of the benchmark datasets. Further detailed features of problem instances of these datasets are summarized in Chen et al. (2021). Table 8 compares these datasets in terms of their constraints. The ITC2019 benchmark dataset includes distribution constraints.[7] The breakdown of the ITC2019 constraints is presented in Müller et al. (2018) and Lemos et al. (2021).

3.4.2. Real-world

Real-world datasets are often gathered by different faculties and institutions. These datasets aim to address the unique needs of a real-world problem. Real-world datasets are of vital importance as they can highlight the gap between literature and real-world implications. Furthermore, they provide a basis for assessing the performance of state-of-the-art approaches in real-world applications. Table 11 overviews the papers addressing real-world datasets in the literature.

3.5. Performance measurements

Aspects to consider while assessing the performance of a method in the UCTP are quality, feasibility, and speed (Chen et al., 2021).

Considering soft and hard constraints, the cost (C) of a candidate solution (S) to be minimized can be measured as $C_S = \sum_{i=1}^{n} W_i SC_i + \sum_{j=1}^{m} W_j SC_j$, where W_i and W_j are the weights associated with each soft and hard constraint violation, respectively. To simplify, constraints are given equal weights, and the cost is often measured as the weighted sum of soft and hard constraint violations count in the literature ($C_S = W_{SC}|SC| + W_{HC}|HC|$). As an infeasible solution is deemed worthless both in benchmarks and real-world applications (Chen et al., 2021), the quality of a candidate solution can be assessed in terms of the number of soft constraint violations.

Another major factor in performance evaluation is speed. Methods should be examined under equal implementation and run conditions

[1] http://sferics.idsia.ch/Files/ttcomp2002/ Last accessed: Feb 01, 2022.
[2] http://www.cs.qub.ac.uk/itc2007/index.htm Last accessed: Feb 01, 2022.
[3] https://www.unitime.org/ Last accessed: Feb 01, 2022.
[4] https://www.itc2019.org/home Last accessed: Feb 01, 2022.
[5] https://iridia.ulb.ac.be/supp/IridiaSupp2002-001/index.html Last accessed: Feb 01, 2022.
[6] http://www.rhydlewis.eu/hardTT/ Last accessed: Feb 01, 2022.
[7] https://www.itc2019.org/format Last accessed: Feb 01, 2022.

Table 7
Features of benchmark datasets.

Dataset		# of Instances	# of Events	# of Rooms	# of Features	# of Students
ITC2002		20	350, 400, 440	10, 11	5, 6, 10	200, 220, 250, 300, 350
ITC2007	Track2	24	100, 200, 300, 400, 500, 600	10, 20	10, 20, 30	300, 500, 1000
Socha	Small	5	100	5	5	80
	Medium	5	400	10	5	200
	Large	1	400	10	10	400
Hard	Small	20	200, 210, 220, 225	5, 6	3, 4, 5, 8, 10	200, 400, 500, 800, 900, 1000
	Medium	20	390, 400, 410, 425	8, 10, 11	5, 6, 8, 9, 10	400, 450, 500, 800, 1000
	Big	20	1000, 1050, 1075	25, 26, 28	10, 20, 25	800, 900, 1000, 1100
Dataset		# of Instances	# of Courses	# of Rooms	# of Curricula	# of Constraints
ITC2007	Track3	21	30 - 131	5 - 20	13 - 150	53 - 1368
Dataset		# of Instances	# of Courses	# of Rooms	# of Classes	# of Students
ITC2019		30	36 - 2839	18 - 768	417 - 8813	0 - 38437

Table 8
Constraints of benchmark datasets.

	Hard Constraints									Soft Constraints						
	HC1	HC2	HC3	HC4	HC5	HC6	HC7	HC8	HC9	SC1	SC2	SC3	SC4	SC5	SC6	SC7
ITC2002	✓	✓	✓	✓						✓	✓	✓				
ITC2007-Track2	✓	✓	✓	✓	✓	✓				✓	✓	✓				
ITC2007-Track3				✓			✓	✓	✓				✓	✓	✓	✓
Socha	✓	✓	✓	✓						✓	✓	✓				
Hard	✓	✓	✓	✓												

for a fair comparison. This can be achieved by using a host computer or taking your system's hardware configuration into consideration (as directed in ITC2007). For clarity, it is customary in the literature to state the programming language and the system's specifications (CPU and RAM) used for benchmarking.

Unlike deterministic methods, the performance of stochastic methods depends on a set of random variables generated (Bianchi et al., 2009). The experimental results of these methods are often reported as an average of several independent runs of the search algorithm to produce more stable results and allow for statistical comparisons to be made (Kesur, 2013).

3.6. Approaches

Approaches addressing the UCTP in the literature can be divided into five main categories (Chen et al., 2021, Babaei et al., 2015): Operational Research (OR) based, meta-heuristics, hyper-heuristics, multi-objective, and hybrid approaches.

Approaches for the UCTP can also be categorized based on their number of steps in addressing the constraints into single and multi-stage (and multi-stage with relaxation) (Lewis, 2008). While single-stage approaches attempt to find solutions satisfying both hard and soft constraints simultaneously, multi-stage approaches tackle hard and soft constraints in different stages.

4. Meta-heuristic approaches in the UCTP

Meta-heuristic (metaheuristic) is defined as "an iterative process guiding heuristics to explore and exploit the search space to find near-optimal solutions" (Osman & Kelly, 1997). Heuristics are approximate approaches that seek a good solution at a reasonable computation cost without the guarantee of finding the optimal solution (Burke & Kendall, 2014). Meta-heuristics operate on a higher level than heuristics (but lower than hyper-heuristics), and they can provide a good solution to an optimization problem under incomplete or imperfect information or limited computation capacity (Bianchi et al., 2009). These general problem solvers are capable of searching a large solution space and handling a variety of different problems as they make relatively few assumptions about the problem (Blum & Roli, 2003). Meta-heuristics can be categorized into single solution-based (often known as Local Search (LS) algorithms) and population-based approaches (Chen et al., 2021, Babaei et al., 2015, Bashab et al., 2020). Fig. 4 presents the categorization of all the meta-heuristic approaches applied to the UCTP.

4.1. Single solution-based approaches

4.1.1. Simulated annealing

Simulated Annealing (SA) is among the best LS algorithms, i.e., heuristic mechanisms to find approximate solutions by considering neighboring solutions (Burke & Kendall, 2014), to tackle COP problems due to their high performance and wide applicability (Burke & Kendall, 2014). Inspired by the analogy of the physical annealing process of solids, the SA concept was introduced in Kirkpatrick et al. (1983) and Černý (1985).

Bellio et al. applied a single-stage SA to artificially-generated problem instances of the CB-CTP (Bellio et al., 2016). To determine the relationship between method parameters and problem instance features, they conducted a statistical analysis. Using cross-validation, method parameters were tuned on the artificial instances. Then, ITC2007 instances were used as validation. And for test instances, they introduced a novel real-world dataset to evaluate the performance of their method. Feature-based tuned SA outperformed the results in the literature on 10 instances out of 21 of the ITC2007-Track3 dataset.

In related research (Song et al., 2018), a multi-stage SA-based Iterated Local Search (ILS) procedure was proposed for the Hard instances introduced in Lewis and Paechter (2007). In the first phase (Initialization), they incorporated a greedy heuristic to produce partial-feasible solutions. Then, SA was employed in the second phase (Intensification) until the local optimum was reached. To further improve the performance in this stage, acceptance of a worse solution and a novel cooling scheme were adopted. In the final phase (Diversification), an improvement-perturbation mechanism was applied to improve or perturb the current solution. This approach managed to find feasible solutions for 58 of the instances out of 60, which is 3 more than previous

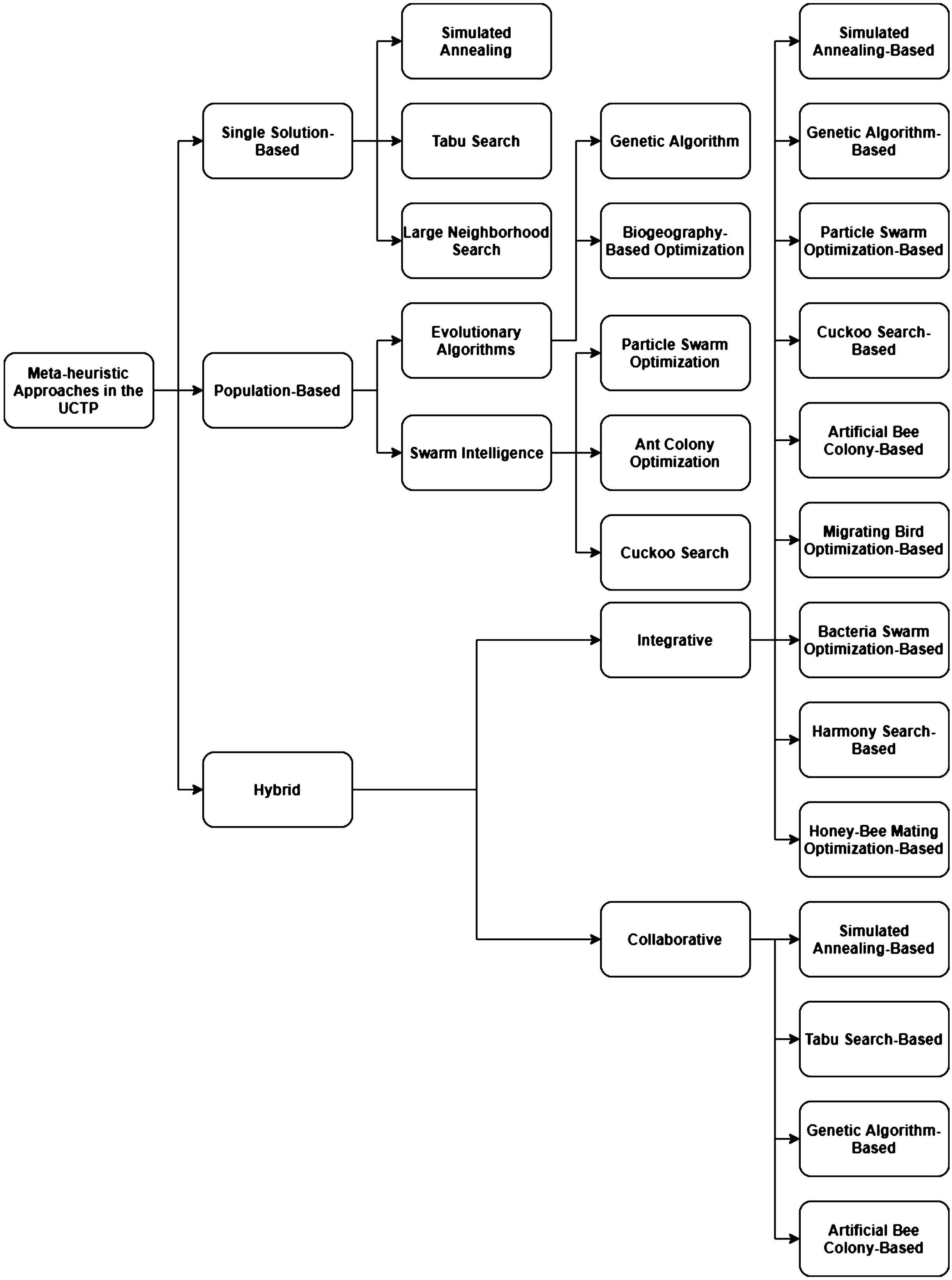

Fig. 4. Meta-heuristic approach categories in the UCTP.

state-of-the-art methods. Furthermore, it achieved better average solution quality and a lower number of unallocated events.

University course timetables are often finalized in stages. Changes are inevitable between these stages, which makes the previous timetable infeasible. Two different variants of these disruptions were investigated in Gülcü and Akkan (2020): single disruption and multiple disruptions. They proposed Multi-Objective Simulated Annealing for Single Disruption (MOSA-SD) and Multi-Objective Simulated Annealing for multiple disruptions with Sample Average Approximation (MOSA-SAA) to address these two problems, respectively. The main difference between these two methods is in how the robustness of a solution is measured. The ITC2007-Track3 benchmark dataset was used for performance evaluation. In the single disruption case, MOSA-SD outperformed the multi-objective genetic algorithm presented in Akkan and Gülcü (2018) in terms of generational distance (see Van Veldhuizen, 1999) and hypervolume (see Zitzler & Thiele, 1999). It also provided a wider range of Pareto optimal solutions. In multiple disruptions, MOSA-SAA outperformed MOSA-SD.

Related research (Akkan et al., 2022) aimed to find resilient timetables that can cope with potential data disruptions, such as changes in the availability of professors or rooms. They modeled the CB-CTP as a bi-criteria optimization problem, where robustness is a stochastic objective, and the objective is to find a good approximation of the Pareto frontier. They developed a Multi-Objective Simulated Annealing (MOSA) algorithm that uses a surrogate measure to estimate the robustness objective. They used ten different slack measures and thirty surrogate measures, inspired by the concept of slack in machine and project scheduling. They tested their method on ITC2007 instances and compared it with other existing methods. They discovered that one of their surrogate measures, when used in a multi-start MOSA algorithm, consistently produced the best Pareto frontier. However, their method still needed manual adjustment of some parameters and did not take into account student preferences or satisfaction in the timetabling process.

A cooperative variant of SA for the UCTP, named Simulated Annealing with Cooperative Processes (SACP), was proposed by Cruz-Rosales et al. (2022). This method employs multiple processes that perform SA on distinct solutions and communicate via collective and point-to-point messages. The collective messages allow the master process to share the best solution among all the processes to explore the best solution. Meanwhile, the point-to-point messages direct the search procedure toward a more promising solution space. SACP was tested on a set of synthetic instances introduced by Rossi-Doria et al. (2003) and outperformed five other basic meta-heuristics according to statistical analysis. However, the method lacked comparison with other state-of-the-art methods and validation on other benchmark datasets.

4.1.2. Tabu Search

Tabu Search (TS) is yet another LS-based meta-heuristic that has been successfully applied to countless COPs. It was first proposed in Glover (1986) and then formalized in Glover (1989, 1990). This approach helps hill climbing overcome local optimum by introducing short and long-term memory. The term tabu refers to preventive measures that stop the algorithm from cycling when moving away from the local optimum through non-improving moves (Burke & Kendall, 2014). The balance between exploitation and exploration in TS can be obtained by employing freeze restart intensification and restart diversification (Burke & Kendall, 2014).

Finding a feasible solution is essential for course timetabling. The Hard benchmark dataset introduced 60 challenging instances with the sole purpose of finding feasible solutions. The problem was first transformed into one that considers only one hard constraint by Chen et al. (2020). Then, they introduced a single-stage novel Tabu Search algorithm with a Controlled Randomization strategy (TSCR) algorithm to tackle this problem. Two complementary neighborhoods were employed to intensify the search, and a threshold mechanism was adopted for the neighborhood search in TSCR. This method was competitive with the 8 compared algorithms and managed to find feasible solutions for 55 instances. Furthermore, it found feasible solutions for all instances when the time limit was extended to 24 hours.

4.1.3. Large Neighborhood Search

A single-stage Adaptive Large Neighborhood Search (ALNS) was applied to the CB-CTP by Kiefer et al. (2017). This algorithm was based on destroying and repairing large parts of solutions in a repetitive manner. Four features for destroy limit, temperature reheating, infeasible solutions allowance, and repair operators computation times were implemented in ALNS, alongside several destroy and repair operators. ALNS achieved highly competitive results for the ITC2007-Track3 dataset and found 5 new best solutions.

4.2. Population-based approaches

4.2.1. Evolutionary Algorithms

Inspired by nature, Evolutionary Algorithms (EAs) are a group of population-based meta-heuristics based on Darwin's theory of evolution (survival of the fittest) (Eiben et al., 2003). These algorithms have shown profoundly promising performance on a diverse set of optimization problems and are common in the literature. The exploration/exploitation balance in these algorithms is accomplished by recombination and mutation operators. Fig. 5 illustrates the general scheme of EAs.

Genetic Algorithm.

Genetic Algorithm (GA) is the most widely used type of EAs (Eiben et al., 2003). It is based on the principles of natural selection and genetics and was introduced in Fraser (1957).

Many necessary constraints in the real-world UCTP are not accounted for in the benchmark datasets. Related research (Abdelhalim & El Khayat, 2016) introduced a new variant of the UCTP with maximizing resource utilization as their objective and proposed a Utilization-based Genetic Algorithm (UGA) to tackle this problem. The novelty of this work was the inclusion of professors' preferences and constraints. Applying to the real-world dataset from the Faculty of Commerce, Alexandria University in Egypt, UGA enhanced the occupancy rates of the allocated events and managed to save resources. However, it was more computationally expensive on smaller instances compared to other methods.

Energy consumption is a big concern for universities. Saving energy can be fulfilled by an efficient allocation of classrooms. However, there have been few attempts to consider spatial and functional capacities related to energy use in classrooms. Song et al. studied the correlation between timetabling and energy usage at the Liberal Arts Building 1 in the Seoul National University campus in Seoul, South Korea (Song et al., 2017). They introduced a new variant of the UCTP, focusing on minimizing energy consumption, and applied a single-stage genetic algorithm to address this problem. This approach contributed to 4% energy saving (up to 5% by discarding the hard constraints).

Current generic solutions do not meet certain specific constraints of the real-world UCTP. A real-world UCTP at Telkom University was addressed in Gozali and Fujimura (2018). A Reinforced asynchronous Island Model Genetic Algorithm (RIMGA) was proposed to optimize the usage of the computer's resources. In this design, the slave islands that had completed their processes were utilized to assist those who had not. RIMGA managed to achieve comparable results with Asynchronous Island Model Genetic Algorithm (AIMGA) in half the time. It was also less likely to get trapped in the local optimum.

In student sectioning UCTP, a set of preferred classes are chosen by students, and then a timetable is created while attempting to minimize constraint violations and adopt students' preferences. To address this problem, a Localized Island Model Genetic Algorithm with Dual Dynamic Migration Policy (DM-LIMGA) was proposed in

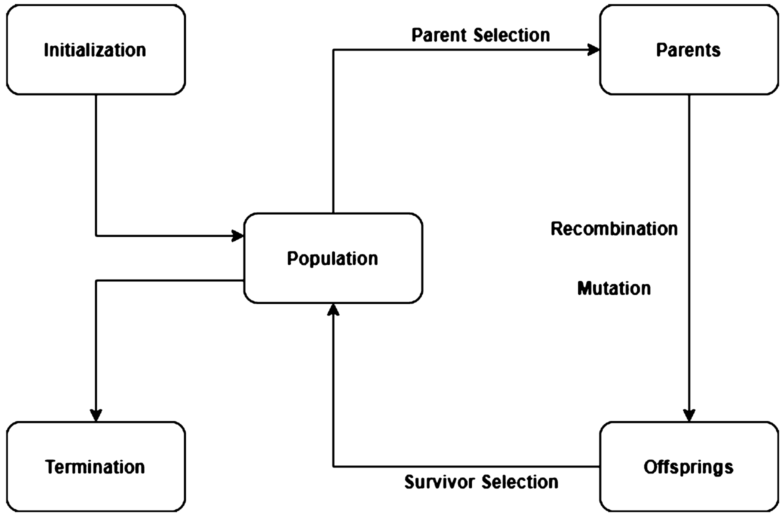

Fig. 5. EAs general scheme (Eiben et al., 2003).

Gozali et al. (2020). In this method, direct representation encoding was used for chromosomes, and each gene block consisted of time, room, lecturer, class, and students. This approach strictly dedicated one slave island to finding feasible solutions, while the second one attempted to minimize soft constraints, and the third one focused on student-level constraints. For each of these islands, a different variant of GA was used. The diversity of each island was estimated using a bias value. DM-LIMGA managed to find feasible solutions for student sectioning UCTP and outperform GA, AIMGA, and UniTime on the Telkom University and ITC2007-Track2 datasets.

Biogeography-Based Optimization.

Biogeography is the study of the geographical distribution of biological organisms. Biogeography-Based Optimization (BBO) is an evolutionary-based, stochastic, iterative optimization method that was first introduced in Simon (2008).

Related research (Zhang et al., 2017) introduced a novel, discrete Ecogeography-Based Optimization (EBO) method to address the Unconstrained University Course Timetabling Problem (UCTP). EBO enhanced BBO by introducing a neighborhood structure for the population. In this work, two local and global operators, along with a repair mechanism, were incorporated to effectively explore the solution space and reduce computational cost. EBO showed competitive performance compared to the state-of-the-art approaches when applied to a set of problem instances from four universities in China. The main limitation of this approach is the need for manual setting of migration rates and the immaturity index parameters.

4.2.2. Swarm Intelligence

Inspired by the collective behavior of swarms or insect colonies, Swarm Intelligence (SI) aims to design and study efficient computational methods for solving problems (Burke & Kendall, 2014, Bonabeau et al., 1999). SI was introduced in Beni and Wang (1993). Though seemingly, there is no evolution in SI, it does fit the general EA framework algorithmically and can be categorized under EA (Eiben et al., 2003). SI methods have found an increasing number of applications in the last few years.

Particle Swarm Optimization.

Particle Swarm Optimization (PSO) is based on the social behavior of bird flocking or fish schooling and was first presented in Kennedy and Eberhart (1995). The core idea of PSO is to consider a point in space with a position and a velocity as a member of a population, where the current velocity determines the new position (and velocity) (Eiben et al., 2003).

Hossain et al. proposed a novel single-stage PSO-based method to tackle the UCTP for a real-world dataset (Hossain et al., 2019). Particle Swarm Optimization with Selective Search (PSOSS) employed a swap sequence-based discrete PSO, in which the velocity was controlled by a sequence for global best and a combination sequence. PSOSS managed to outperform GA and HS on the dataset from the Computer Science and Engineering Department of Khulna University of Engineering and Technology.

A Particle Swarm Optimization Based Timetabling (PSOT) tool was presented in Thepphakorn and Pongcharoen (2019). In this single-stage approach, the conventional Particle Swarm Optimization (PSO), the Standard PSO (SPSO), and the Maurice Clerc PSO (MCPSO) were implemented. Applying this tool to the 5 real-world datasets collected from their previous work (Thepphakorn et al., 2016), MCPSO outperformed the other variants of PSO for most datasets. Moreover, through conducting a statistical experiment, it was found that the setting of PSOs' parameters was significant with a 95% confidence interval.

Ant Colony Optimization.

Ant Colony Optimization (ACO) algorithm is based on the pheromone-based communication of ants. It was inspired by the double-bridge experiment in Colorni et al. (1991). ACO algorithms have been designed and successfully applied to many different types of COPs, including dynamic and multi-objective optimization problems (Burke & Kendall, 2014).

Student grouping (placing students in disjoint groups where each student belongs to exactly one group based on selected events) was investigated in Badoni and Gupta (2015b). Then, a single-stage ACO algorithm based on student grouping was presented and applied to 11 instances obtained from the Socha dataset. Ant Colony Optimization With Student Groupings (ACOWSG) excluded students from further selection once they were assigned a group. ACOWGS managed to outperform ACO on all the studied instances and was competitive with 9 other methods on 9 of the instances.

A single-stage ACO to tackle CB-CTP was proposed in Kenekayoro and Zipamone (2016). Unlike other ACO-based studies that incorporate different local search algorithms for the improvement phase, an ant system was used here. The proposed approach was able to find feasible solutions for all the ITC2007-Track3 instances and near-optimal solutions for some instances. The main drawback of this approach is the high computational time of the improvement phase.

Cuckoo Search Algorithm.

Cuckoo Search (CS) is yet another novel SI-based approach. CS is based on the aggressive brood parasitism of some cuckoo species and

Table 9
Summary of approaches on different UCTP datasets.

UCTP	Meta-heuristic		Hybrid Meta-heuristic		Total
	Single Solution-Based	Population-Based	Collaborative	Integrative	
Benchmark	7	2	7	15	31
Real-World	0	8	0	6	14
Total	7	10	7	21	45

Table 10
Categorization of approaches applied to different UCTP benchmark datasets.

Dataset	Year	Methodology	Category	Reference
ITC2002	2017	Simulated Annealing	Hybrid (Collaborative)	Goh et al. (2017)
	2019	Simulated Annealing	Hybrid (Collaborative)	Goh et al. (2019b)
	2020	Tabu Search	Hybrid (Collaborative)	Goh et al. (2020)
ITC2007-Track2	2015	Simulated Annealing	Hybrid (Collaborative)	Lewis and Thompson (2015)
	2015	Genetic Algorithm	Hybrid (Collaborative)	Soria-Alcaraz et al. (2015)
	2017	Simulated Annealing	Hybrid (Collaborative)	Goh et al. (2017)
	2019	Simulated Annealing	Hybrid (Collaborative)	Goh et al. (2019b)
	2020	Tabu Search	Hybrid (Collaborative)	Goh et al. (2020)
	2020	Genetic Algorithm	Population-Based	Gozali et al. (2020)
	2021	Genetic Algorithm	Hybrid (Integrative)	Rezaeipanah et al. (2021)
ITC2007-Track3	2015	Harmony Search Algorithm	Hybrid (Integrative)	Wahid and Hussin (2015)
	2016	Simulated Annealing	Single Solution-Based	Bellio et al. (2016)
	2016	Ant Colony Optimization	Population-Based	Kenekayoro and Zipamone (2016)
	2016	Genetic Algorithm	Hybrid (Integrative)	Yousef et al. (2016)
	2017	Large Neighborhood Search	Single Solution-Based	Kiefer et al. (2017)
	2018	Genetic Algorithm	Hybrid (Integrative)	Akkan and Gülcü (2018)
	2018	Genetic Algorithm	Hybrid (Integrative)	Matias et al. (2018b)
	2020	Simulated Annealing	Single Solution-Based	Gülcü and Akkan (2020)
	2021	Simulated Annealing	Hybrid (Integrative)	Song et al. (2021)
	2022	Simulated Annealing	Single Solution-Based	Akkan et al. (2022)
Socha	2015	Ant Colony Optimization	Population-Based	Badoni and Gupta (2015b)
	2015	Artificial Bee Colony	Hybrid (Collaborative)	Ghasemi et al. (2015)
	2015	Artificial Bee Colony	Hybrid (Integrative)	Fong et al. (2015)
	2015	Migrating Bird Optimization	Hybrid (Integrative)	Shen et al. (2015)
	2015	Bacteria Swarm Optimization	Hybrid (Integrative)	Shaker et al. (2015)
	2017	Simulated Annealing	Hybrid (Collaborative)	Goh et al. (2017)
	2017	Honey-Bee Mating Optimization	Hybrid (Integrative)	Aziz et al. (2017)
	2019	Simulated Annealing	Hybrid (Collaborative)	Goh et al. (2019b)
	2020	Tabu Search	Hybrid (Collaborative)	Goh et al. (2020)
Hard	2018	Simulated Annealing	Single Solution-Based	Song et al. (2018)
	2020	Tabu Search	Hybrid (Collaborative)	Goh et al. (2020)
	2020	Tabu Search	Single Solution-Based	Chen et al. (2020)

their egg-laying strategy. CS was first developed and introduced in Yang and Deb (2009).

With the high number of conflicting objectives in the UCTP, a weight sum approach (adopting a single objective by combining criteria) might be infeasible. Thepphakorn et al. proposed a Multi-Objective Cuckoo Search based Timetabling (MOCST) tool to address the multi-objective UCTP for minimizing the total operating costs and the number of inadequate chairs (Thepphakorn et al., 2016). The CS via Lévy Flight (CSLF) and CS via Gaussian Random Walk (CSGRW) were embedded in MOCST to find the Pareto optimal solutions. Applying MOCST to the 11 datasets obtained from Naresuan University in Thailand, CSLF outperformed CSGRW for almost all datasets.

4.3. Discussion

Table 9 summarizes the approaches used to address different UCTP datasets. Out of the 45 papers surveyed, 17 were meta-heuristics, including 7 single solution-based and 10 population-based methods. All of the single solution-based approaches were applied to benchmark datasets (see Table 9). This may be because these methods have higher exploitative ability than population-based methods (Du et al., 2016), which allows them to find high-quality solutions under a strict time constraint. However, population-based methods have better exploratory ability (Du et al., 2016), and they were more prevalent in the literature (10 versus 7) than single solution-based methods, as shown in Table 9.

Table 10 and 11 classify the methodologies based on the benchmark and real-world datasets they used, respectively. Most of the population-based methods (8 out of 10) were applied to real-world datasets and achieved promising results (see Table 11). However, their application to benchmark datasets was relatively scarce (only 2 in our survey), and they did not perform as well as single solution-based methods in the ITC competitions (see Table 6). This is due to their higher time-complexity trade-off.

A major drawback of meta-heuristics is the need for parameter setting. The performance of meta-heuristics can be greatly influenced by the parameter settings (Thepphakorn & Pongcharoen, 2019, Rodríguez Maya et al., 2016, Thepphakorn et al., 2021). To address this issue, the irace package was proposed in López-Ibáñez et al. (2016) to find the best parameter settings for an optimizer. However, there is still a lack of analysis and strategies for parameter control/tuning in the literature.

Among the 7 single solution-based approaches reviewed, 5 were based on simulated annealing, 1 on tabu search, and 1 on large neighborhood search (see Table 12). Simulated annealing has shown remarkable capabilities in solving UCTP benchmark datasets (see Table 6) and

Table 11
Categorization of approaches applied to different real-world UCTP datasets.

Year	Methodology	Category	Faculty	University	Country	Reference
2016	Genetic Algorithm	Population-Based	Faculty of Commerce	Alexandria University	Egypt	Abdelhalim and El Khayat (2016)
2016	Cuckoo Search	Population-Based	Faculty of Engineering	Naresuan University	Thailand	Thepphakorn et al. (2016)
2017	Genetic Algorithm	Population-Based	Case Building	Seoul National University	South Korea	Song et al. (2017)
2017	Biogeography-Based Optimization	Population-Based		4 Universities	China	Zhang et al. (2017)
2017	Harmony Search Algorithm	Hybrid (Integrative)	College of Arts and Sciences	Universiti Utara Malaysia	Malaysia	Wahid and Mohd Hussin (2017)
2018	Genetic Algorithm	Population-Based	Engineering	Telkom University	Indonesia	Gozali and Fujimura (2018)
2018	Genetic Algorithm	Hybrid (Integrative)	Department of Information Technology Education	Caraga State University	Philippines	Matias et al. (2018a)
2019	Particle Swarm Optimization	Population-Based	Computer Science and Engineering Department	Khulna University	Bangladesh	Hossain et al. (2019)
2019	Particle Swarm Optimization	Population-Based	Faculty of Engineering	Naresuan University	Thailand	Thepphakorn and Pongcharoen (2019)
2020	Cuckoo Search	Hybrid (Integrative)	Faculty of Engineering	Naresuan University	Thailand	Thepphakorn and Pongcharoen (2020)
2020	Particle Swarm Optimization	Hybrid (Integrative)	Faculty of Engineering	Naresuan University	Thailand	Thepphakorn et al. (2020)
2021	Particle Swarm Optimization	Hybrid (Integrative)	Faculty of Engineering	Naresuan University	Thailand	Thepphakorn et al. (2021)
2022	Genetic Algorithm	Hybrid (Integrative)	International Campus	Universiti Malaysia Sabah	Malaysia	Wong et al. (2022)

appears to be the best-performing single solution-based method. Tabu search has also proven to be highly effective in minimizing the hard constraint violations (Chen et al., 2020).

As shown in Table 9, from the 10 population-based approaches, 5 were EA-based and 5 were SI-based. Among the EA-based approaches, 4 were genetic algorithms and 1 was EBO (see Table 12). GA-based approaches have been the most common techniques in our survey, mainly because of their high flexibility when applied to different problem instances (Song et al., 2017, Gozali et al., 2020). However, they have not always succeeded in finding feasible solutions under strict time constraints (Abdelhalim & El Khayat, 2016). The 5 SI-based approaches reviewed in our survey included 2 PSO, 2 ACO, and 1 CS approach (see Table 12).

5. Hybrid meta-heuristics in the UCTP

Hybrid approaches combine two or more different methods to provide more efficient and flexible solutions for real-world and large-scale problems (Blum et al., 2008). The main goal of using hybridization techniques is to achieve high-quality solutions by striking an optimal balance between global and local search during the optimization process (Shirvani, 2020, Noorian Talouki et al., 2021, Tanha et al., 2021). Hybridization has led to good results in previous research (Chen et al., 2021, Babaei et al., 2015, Bashab et al., 2020). In general, hybrid approaches can be classified as either collaborative combinations or integrative combinations (Blum et al., 2008). Collaborative (cooperative) hybrid approaches exchange information (sequentially, intertwined, or in parallel) but are not part of each other, while in integrative hybrid approaches, one technique is an embedded component of another technique (Blum et al., 2008, Delorme et al., 2010). Table 12 presents a comprehensive summary of all the (hybrid) meta-heuristic approaches studied in this survey paper.

5.1. Collaborative approaches

5.1.1. Simulated annealing

A collaborative multi-stage approach based on SA to address PE-CTP was proposed in Lewis and Thompson (2015). The first stage used the PARTIALCOL algorithm to find feasible solutions by minimizing the Distance To Feasibility (DTF) measurement. Then, SA was employed to explore the space of feasible solutions by minimizing the soft constraint violations. The proposed method outperformed the literature on the ITC2007-Track2 dataset. A further contribution of this work was the study of the effect of solution space connectivity on solution quality. It was shown that higher solution connectivity generally leads to higher-quality solutions. It would be interesting to see how other neighborhood operators would affect the solution space connectivity.

A challenging issue in SA (and meta-heuristics in general) is the extensive parameter tuning that is often required. Related research (Goh et al., 2017) addressed this issue for PE-CTP by presenting a collaborative hybrid approach. In the first stage, Tabu Search with Sampling and Perturbation (TSSP) was used to find feasible solutions. To further improve the quality of solutions, in the second stage, an improved version of simulated annealing, called Simulated Annealing with Reheating (SAR), was proposed. This method introduced self-adaptive tuning of the temperature parameter based on the balance of exploration and exploitation. However, the neighborhood structure still had to be set manually. TSSP was highly effective and achieved 100% feasibility on Socha, ITC2002, and ITC2007 datasets. Furthermore, SAR was comparable to other state-of-the-art approaches in reducing soft constraint violations.

To address the shortcoming of their previous work, a reinforcement learning-based composition of neighborhood structure was incorporated in SAR to create Simulated Annealing with Improved Reheating and Learning (SAIRL) in order to further improve solution quality for PE-CTP (Goh et al., 2019b). This eliminated the need for manual setting of neighborhood structure in SAR. Finding feasible solutions was handled identically by using TSSP in the first stage. SAIRL was highly competitive with SAR and TSSP + SAIRL achieved new best results for 6 instances and new mean results for 14 instances on the Socha, ITC2002, and ITC2007-Track2 benchmarks.

5.1.2. Tabu Search

A further extension of Goh et al. (2017) was presented in Goh et al. (2020). Here, TSSP was hybridized with ILS in an integrative manner. If TSSP failed to find a feasible solution, the best-found solution was passed to an iterative local search in the last quarter of the execution time for further improvement. TSSP-ILS outperformed both TSSP and ILS in finding feasible solutions for stage 1. Moreover, it did not require manual parameter setting, which made it a leading approach for finding feasible solutions. As SAR required a manual setting of neighborhood structure, two preliminary runs were added to it so that a good composition could be obtained automatically. Tabu Search with Sampling and Perturbation with Iterated Local Search + Simulated Annealing with Reheating with Two Preliminary runs (TSSP-ILS + SAR-2P) achieved new best results for 3 instances and new best mean results for 7 instances when applied to Socha, ITC2002, and ITC2007-Track2 datasets in addressing PE-CTP.

5.1.3. Genetic Algorithm

The corresponding study (Rezaeipanah et al., 2019) proposed a GA-based collaborative hybrid approach to tackle UCTP. Parallel Genetic Algorithm and Local Search (PGALS) used a direct representation of a timetable and encoded the distance to feasibility measurement in the fitness function to prevent the generation of infeasible solutions. After the termination condition of GA was met, an LS with a maximum number of iterations was applied to the best chromosome to improve the quality of the solution. When applied to the BenPaechter dataset, the proposed algorithm produced some of the best-known results but was unable to find feasible solutions for all large instances.

With the technological advancements in multi-core and hyper-threading technologies, the solution quality and the Number of Fitness Evaluations (NFE) needed for parallel design of heuristics can greatly benefit compared to conventional sequential approaches. Soria et al. implemented and investigated a parallel set of heuristic algorithms based on GAs, Scatter Search (SS), and discrete PSO for PE-CTP (Soria-Alcaraz et al., 2015). A further contribution of this work was the introduction of "Methodology of Design" which ensures easy adaptability to new instances in order to improve generality. Conducting 100 independent comparative runs between sequential and parallel computing models for GA, SS, and PSO, cGA demonstrated high potential in terms of solution quality and speed.

5.1.4. Artificial Bee Colony

Artificial Bee Colony (ABC) is an SI-based optimization algorithm. It is inspired by a particular intelligent behavior of honey bee swarms and was first introduced in Karaboga (2005).

A multi-stage collaborative approach was presented in Ghasemi et al. (2015) based on an ABC algorithm. In the first stage, Genetic Grouping (GG) was employed to generate feasible solutions. These solutions were then passed to an ABC algorithm to minimize the soft constraint violations. A novel neighborhood structure based on three neighborhoods was applied to both stages. The proposed approach was applied to medium and large instances of the Socha dataset and achieved better performance in 4 out of 5 cases compared with 3 other hybrid methods.

5.2. Integrative approaches

5.2.1. Simulated annealing

A novel Competition-guided Multi-neighborhood Local Search (CMLS) algorithm based on SA was proposed in Song et al. (2021) to tackle CB-CTP. In the first stage of this multi-stage approach, a greedy heuristic was used to generate a feasible solution. Then, six neighborhood operators were adopted in the proposed SA-based multi-neighborhood local search. Here, a new way of combining multiple neighborhoods was presented. To determine the probabilities of neighborhood selection, two heuristic rules were proposed. Finally, the elite solution was chosen for the next iteration from the two SA procedures, each with a different probability set, through the competition-based restart strategy. This approach achieved 16 best average results for the ITC2007-Track3 dataset. The main limitation of this approach is the need for manual setting of the selection probabilities of the different neighborhoods, which can be addressed by an adaptive method in future research.

5.2.2. Genetic Algorithm

A single-stage Hybrid approach combining a steady-state Genetic Algorithm with a Local Search technique and Tabu Search (HGALTS) was presented in Jaengchuea and Lohpetch (2015). LS and TS were integrated into the procedure of GA to address PE-CTP. An LS, based on three neighborhoods, was applied to the initial random population and offspring after crossover and mutation. The quality of offspring was further improved by applying TS. HGALTS managed to find feasible solutions for all 11 instances of the "MN dataset" (Socha et al., 2002) and was competitive with 16 other methods from the literature.

A combination of Genetic Algorithm and Iterated Local Search (GAILS) was employed by Badoni and Gupta (2015a) to tackle UCTP. GAILS took advantage of the diversification ability of GA and intensification superiority of ILS for fast convergence and avoiding local optimum. The ILS employed three neighborhoods and four perturbation moves. It was applied to individuals after random initialization and mutation during the GA process. GAILS was able to find the optimal solutions for all small instances of the dataset adopted from Rossi-Doria et al. (2002) and new best results for two of the medium instances.

Yousef et al. presented an integrative hybrid GPU-based Genetic Algorithm (Yousef et al., 2016). In this parallel, single-stage approach, GA was employed to address CB-CTP. Gender selection was utilized to balance selection pressure and keep a diverse population. Moreover, LS was applied to each produced offspring after crossover and mutation. The fitness function was parallelized, using the CUDA framework, and was GPU accelerated. For large instances of the ITC2007-Track3 dataset, this approach achieved up to 2.8 times faster time.

Addressing the multi-objective PE-CTP in Lohpetch and Jaengchuea (2016), a Hybrid Non-Dominated Sorting Genetic Algorithm-II with Two LS techniques and a TS heuristic (HNSGA2LTS) approach was suggested. TS and LS approaches were applied to child solutions after the crossover and mutation operators. This approach was tested on the MN dataset, and it was shown that the embedded TS and LS approaches helped improve the exploration ability of the NSGA-II, while the introduced LS approach took the role of improving solution quality. Moreover, the final produced result was a set of non-dominated solutions, which gave the users the opportunity to select the most preferable solution from the set of non-dominated solutions.

Feng et al. extended the standard UCTP by incorporating consecutiveness and periodicity conditions of multi-session lectures as decision variables, which are common, realistic conditions observed in many Eastern Asian universities (Feng et al., 2017). Then, they presented an integrative Hybrid Genetic Algorithm (HGA) and Mixed Integer Linear Programming (MILP) to address this UCTP. A Layer-based Bottom Deepest Left with Fill (LBDLF) strategy was employed for the assignment of lectures. The problem was converted into a three-dimensional container packing problem (3DCPP). Then, MILP and HGA with an embedded LS were utilized to solve this problem. HGA outperformed TS in terms of solution quality for the small, medium, and large instances adopted from the ITC-2007 benchmark dataset.

Four neighborhood structures were integrated into GA in Matias et al. (2018a) to address a real-world UCTP. After the random population initialization, individuals were evaluated, and feasible solutions were collected. A guided repair mechanism was introduced and applied to infeasible timetables. After crossover and mutation, a neighborhood operator was selected and applied. In this approach, a data structure, keeping track of the least used resources, was maintained as a guided or directed strategy to improve the previously generated individuals. The performance of the method was evaluated on a real-world dataset from the Department of Information Technology Education at Caraga State University. The proposed methodology outperformed the classical GA in terms of speed and solution quality.

As an extension of their previous work, a GA with guided search and self-adaptive neighborhood strategies was proposed in Matias et al. (2018b). The general procedure of GA, the utilization of a guiding data structure, and the introduction of a repair mechanism remained similar to their previous work. The data structure was used to guide the neighboring structures and the repair operator to utilize unused pairs of rooms and time slots. Furthermore, a self-adaptive mechanism was integrated after the genetic operators to enhance the optimality of individuals. This proposed methodology produced optimal or near-optimal solutions for the instances of the ITC2007-Track3 dataset when compared to the literature.

Changes after the finalization of a timetable are sometimes inevitable. A robust timetable can easily adapt to changing inputs. Akkan et al. considered late changes in an event's time in CB-CTP and introduced an integrative hybrid GA-based approach to undertake this problem (Akkan & Gülcü, 2018). Robustness was introduced as a practical measurement alongside constraint violations, and a Multi-Objective Genetic Algorithm hybridized with Hill Climbing and Simulated Annealing (MOGA + HC + SA) was proposed. The fitness of individuals was calculated based on their violations and robustness, with four measurements included in assessing robustness. An HC-based mutation was applied to selected parents to form offspring. And, to further improve the population, SA was randomly applied to individuals in the final stage. The Pareto-fronts resulted from this approach included highly robust solutions while maintaining competitive quality in terms of constraint violations on the ITC2007 Track3 dataset. Moreover, the solutions were widely diverse and provided alternatives.

Related work (Rezaeipanah et al., 2021) proposed a multi-stage integrative hybrid approach based on Parallel GA (PGA). The Improved PGA hybridized with LS (IPGALS) started with creating parallel populations of feasible solutions. Then, LS was applied to GA after crossover and mutation to enhance its performance and prevent it from getting stuck in local optima. A Distance to Feasibility (DF) measurement (overall number of students in conflicting events) was employed as guidance toward feasible solutions. Finally, an elitism approach stored the best individuals in shared memory. IPGALS achieved competitive performance on small and medium data instances compared with state-of-the-art approaches when applied to ITC2007-Track2 and BenPaechter datasets but failed to produce feasible solutions for large instances.

Wong et al. proposed an integrative hybrid GA that incorporates TSSP within the first step of the GA procedure to solve a real-world PE-CTP that arises at Universiti Malaysia Sabah (UMS-LIC) (Wong et al., 2022). The TSSP was utilized in the first step of the GA to generate a pool of feasible solutions satisfying the hard constraints. They conducted experiments to find the optimal parameter values for the GA under a preset computational time limit and tested their method on a real-world dataset collected from the semester 1, session 2018/2019 student registration data. They compared their automated timetables with those manually generated by the administrative staff of UMS-LIC and found that their method reduced hard and soft constraint violations by as much as 54%.

5.2.3. *Particle Swarm Optimization*

Thepphakorn et al. proposed a Hybrid Particle Swarm Optimization-based Tool (HPSOT) that combined Maurice Clerc PSO (MSPSO) with a local search (LS) approach (Thepphakorn et al., 2020). The LS approach consisted of Insertion Operators (IO) and Exchange Operators (EO) that were used to improve the solutions generated by MSPSO. HPSOT was applied to a variant of UCTP that aimed to minimize the total operating costs. Five different combinations of IO and EO were tested in HPSOT. HPSOT outperformed MSPSO on 11 real-world instances from their previous work (Thepphakorn et al., 2016) in terms of operating costs, running time, and convergence speed.

A further improvement to HPSOT was presented in Thepphakorn et al. (2021) by incorporating Standard PSO (SPSO). In this single-stage hybrid approach, two types of LS, namely Insertion Operator (IO) and Exchange Operator (EO), were integrated with PSO, and five different IO:EO ratios were evaluated and compared. A repair mechanism was used to handle infeasible solutions. The hybrid SPSO and MCPSO with IO:EO ratios achieved better average total operating costs than their original versions for all problem instances. Moreover, hybridization showed to improve computational complexity.

An integrative approach called Hybridizing Genetic-based Discrete PSO with LS and TS (HGDPSOLTS) was developed in Unprasertporn and Lohpetch (2020) to solve the PE-CTP. The genetic-based discrete PSO adopted the concepts of GA to PSO by using a population of swarms, crossover, and mutation operators. In this multi-stage approach, LS was used to find feasible solutions first. LS and TS were embedded into HGDPSOLTS and were applied to swarms after crossover and mutation operations. This approach leveraged the exploration ability of GDPSO and the exploitation of LS and TS. HGDPSOLTS was applied to the 11 instances of the MN dataset and outperformed all other approaches in the literature in terms of the number of soft constraint violations.

5.2.4. *Cuckoo Search*

An enhancement to the Cuckoo Search (CS) algorithm was suggested in Thepphakorn and Pongcharoen (2020) utilizing a Self-adaptive Parameter Setting (SPS), a movement strategy based on Lévy flight or Gaussian random walks, and local search hybridization based on Insertion Operator (IO) and Exchange Operator (EO). Hybrid Self-adaptive Cuckoo Search-based timetabling (HSCST) followed 6 steps: initialization, strategic movement procedure, repairing, fitness evaluation, hybridization, and parameter updating. HSCST was tested on the 11 instances of the real-world dataset from the Faculty of Engineering, Naresuan University, and outperformed conventional CS and PSO. Moreover, the hybridization of CS with local search improved the feasibility and total operation cost of the solutions.

5.2.5. *Artificial Bee Colony*

To overcome the limitations of exploration and exploitation capabilities in the literature, Fong et al. proposed an integrative hybrid swarm-based approach to solve both UCTP and UETP (Fong et al., 2015). The hybrid Nelder-Mead Great Deluge Artificial Bee Colony (NMGD-ABC) combined a PSO-based global best model to enhance exploration with Great Deluge (GD) to intensify exploitation. Thus, the proposed method was able to maintain a good balance between exploration and exploitation improving the convergence speed of ABC. NMGD-ABC was applied to the Socha and Carter dataset and significantly outperformed ABC. This was one of the few works in the literature that addressed generality by tackling different problems.

5.2.6. *Migrating Bird Optimization*

Migrating Bird Optimization (MBO) was first proposed in Duman et al. (2011). This SI-based method mimics the v-shaped formation of migrating birds during seasonal changes.

Falling into local optimum has been identified as the main weakness of MBO. Shen et al. attempted to overcome this limitation by proposing a single-stage hybrid approach to solve the PE-CTP (Shen et al., 2015). The proposed Modified Migrating Bird Optimization (M-MBO) algorithm began by creating a random population of feasible solutions generated using a combination of multiple graph coloring heuristics. ILS was integrated within this approach to improve the best solution in the next phase. Then, a neighborhood-sharing mechanism was used to help MBO escape local optimum and improve the quality of non-leading solutions. Comparing basic MBO and the proposed M-MBO on the 11 instances of the Socha dataset, M-MBO produced better quality solutions and performed faster. However, the exploitation ability of M-MBO was still insufficient.

5.2.7. *Bacteria Swarm Optimization*

Bacteria Swarm Optimization (BSO) is another novel SI-based method that was proposed in Shaker et al. (2015). BSO is inspired by the behavior of bacteria searching for nutrients.

Related research (Shaker et al., 2015) integratively incorporated Differential Evolution (DE) algorithm within BSO to solve the UCTP. In this multi-stage approach, a constructive heuristic was used to create an initial population of feasible solutions. Then, the search space was divided into three regions: risk, null, and rich. DE was applied within the BSO procedure to guide the solutions, find the global minimum, improve the convergence, and use fewer control parameters. BSO had a faster convergence speed than other methods from the literature on the Socha dataset.

5.2.8. *Harmony Search Algorithm*

Harmony Search (HS) is a simple yet effective evolutionary algorithm. It simulates the improvisation of music players (especially Jazz musicians) and was proposed in Geem et al. (2001) to solve the Traveling Salesman Problem (TSP).

Integrative hybridization of HS with GD to solve the CB-CTP was proposed in Wahid and Hussin (2015). In the first stage, initial feasible solutions were generated using a constructive heuristic. Then, three hybridizations of GD with HS (GD within the RC operator of HS (NGD), GD within the MC operator (GDN), and GD within both (GDGD)) were proposed and compared. The NGD produced solutions with the lowest total cost among these three versions of hybridization. Moreover, it was able to achieve competitive results with the literature on the instances of the ITC2007-Track3 benchmark dataset.

A further improvement of the previous work was presented in Wahid and Mohd Hussin (2017). Hard constraint violations were handled in a similar way in the first stage of this approach. Then, GD was embedded into the random consideration operator of HS. This approach was applied to the real-world dataset of the College of Arts and Sciences, Universiti Utara Malaysia, which consisted of 247 courses, 850 lectures, 32 rooms, 350 lecturers, and 20,000 students to be scheduled on a five-day week (Sunday to Thursday). Their proposed method outperformed their existing timetabling software.

5.2.9. Honey-Bee Mating Optimization

The Honey-Bee Mating Optimization algorithm was proposed in Haddad et al. (2006). It imitates the behavior of honey bees during mating in nature and uses the crossover and mutation operators of GA.

Steepest descent LS is used as a worker in the standard HBMO. This makes this method susceptible to falling into the local optimum, which affects performance. To overcome this problem, an integrative hybridization of HBMO with Adaptive Guided Variable Neighborhood Search (HBMO-AGVNS) as the worker was investigated by Aziz et al. (2017). In the first stage, AGVNS created a population of feasible solutions. Then, the most suitable neighborhood structure was used to handle the soft constraint violations. HBMO-AGVNS showed a good balance between explore and exploit, and the integration of AGVNS helped in escaping the local optimum. This approach outperformed its individual components and was competitive on the Socha dataset instances.

5.3. Discussion

Hybridization of local search and population-based approaches (also known as the Memetic Algorithm (MA)) has shown remarkable performance in solving the UCTP. The balance of exploration and exploitation in these approaches enables them to explore the solution space effectively. Moreover, the hybridization of different methods can help enhance their performance by combining the strengths of each component and avoiding their weaknesses. As shown in Table 9, 28 out of the total 45 reviewed papers in this survey were hybrid, which further indicates the popularity of these approaches in recent years. 7 of these approaches were hybridized in a collaborative manner, while the remaining 21 used integrative hybridization. All the collaborative approaches and 15 out of the 21 integrative approaches have been tested on benchmark datasets.

Collaborative hybridization of simulated annealing, in particular, has been very successful in producing high-quality solutions on benchmark datasets (see Table 10). The collaborative hybridization of SA and TS proposed in Goh et al. (2020) is among the current state-of-the-art on several benchmark datasets. Advantages of collaborative approaches include simpler implementation due to the independent operation of their components (Blum et al., 2008).

This survey revealed that not only are integrative approaches common in the literature (see Table 9), but they are also capable of handling different UCTP problems efficiently (Table 12). 10 out of the 21 integrative approaches were based on genetic algorithms, as shown in Table 12. Integration of exploitative single solution-based approaches within the genetic operators of a GA has resulted in a good performance on real-world and benchmark datasets (see Tables 10 and 11). Another observation is that many novel, swarm intelligence-based approaches with integrative exploitative strategies were introduced in the literature since 2015. However, further research has not been conducted to extend them (refer to Table 12).

6. Future work

To bridge the gap between real-world UCTP and the literature, the International Timetabling Competition was organized by Practice and Theory of Automated Timetabling (PATAT) and the Metaheuristic Network. The field researchers have also contributed to achieving this goal over the years. However, this gap is still large, which has made many universities rely on manual timetabling. In meta-heuristic approaches, this gap can be attributed to the lack of generality among these methods in the literature. The excessive parameter tuning of meta-heuristics in pursuit of optimality on a specific problem can reduce their general applicability (Zamli, 2018, Bibai et al., 2010). We strongly recommend future research to focus on the gap between the UCTP approaches in the literature and their real-world implications to identify their underlying causes. Future studies can be conducted to introduce measurements to assess the generality of approaches, along with their optimality to discourage problem-tailored solutions and reduce the gap.

Hybridization of approaches seems to be the best-performing approach in the literature. Combining different methods can improve their performance by eliminating the weaknesses of each one and exploiting their strengths (Matias et al., 2018b). There are many research opportunities to explore alternative hybrid meta-heuristics, especially on the latest benchmark dataset (ITC2019).

The main drawback of meta-heuristics is the need for parameter setting. Numerous studies have confirmed the effect of parameter setting on the performance of these approaches (Thepphakorn & Pongcharoen, 2019, Rodríguez Maya et al., 2016, Thepphakorn et al., 2021). Therefore, a suitable set of parameters is essential for optimal performance. Future research can be conducted on different parameter control/tuning techniques. It would be very interesting to observe how different parameter settings can affect the performance of (hybrid) meta-heuristics, both in terms of their optimality and general applicability.

Many recent, successful meta-heuristic approaches such as Bat Algorithm (BA) and Grey Wolf Optimizer (GWO) have never been applied to the UCTP problem in the literature. Future research can investigate the effectiveness of these methods on real-world and benchmark UCTP and compare the results with common meta-heuristic approaches in the literature to identify their strengths and weaknesses.

The operation of many universities has changed significantly since 2020. With the outbreak of COVID-19, numerous universities have switched to virtual learning, and many international students have returned to their home countries. This eliminates many of the constraints of the standard UCTP (such as the maximum number of students in a class, the necessary facilities in a classroom, and the maximum physical distance between consecutive classes) and introduces some new constraints (such as consideration of different time zones). Furthermore, the post-covid operation of universities introduces new challenges (such as adhering to 50% room capacity and maintaining minimal physical interactions). Future research can investigate these changes, introduce appropriate datasets, and address these problems.

7. Conclusion

University course timetabling is a crucial task for many educational institutions. The high number of constraints and the immense size of its solution space have made this challenging task an active and important research area. An accurate scheduler is essential for the efficient operation of universities. Meta-heuristic and hybrid meta-heuristic approaches are widely applied to the UCTP in the literature due to their high flexibility and exploration/exploitation balance. These methods

Table 12
Literature review summary.

Year	Authors	Problem	Algorithm	Approach	Single/Multi-Stage	Dataset
2015	Badoni and Gupta (2015b)	UCTP	Ant Colony Optimization With Student Groupings (ACOWSG)	Swarm Intelligence	Single-stage	Socha
	Lewis and Thompson (2015)	PE-CTP	PARTIALCOL + Simulated Annealing	Hybrid (Collaborative)	Multi-stage	ITC2007-Track2
	Ghasemi et al. (2015)	UCTP	Genetic Grouping + Artificial Bee Colony	Hybrid (Collaborative)	Multi-stage	Socha
	Jaengchuea and Lohpetch (2015)	PE-CTP	Hybrid Genetic Algorithm with Local Search and Tabu Search (HGALTS)	Hybrid (Integrative)	Single-stage	MN
	Badoni and Gupta (2015a)	UCTP	hybrid Genetic Algorithm with Iterated Local Search (GAILS)	Hybrid (Integrative)	Single-stage	Rossi-Doria
	Fong et al. (2015)	UCTP	Nelder-Mead Great Deluge Artificial Bee Colony (NMGD-ABC)	Hybrid (Integrative)	Single-stage	Socha, Carter's
	Shen et al. (2015)	PE-CTP	Modified Migrating Bird Optimization (M-MBO)	Hybrid (Integrative)	Multi-stage	Socha
	Shaker et al. (2015)	UCTP	Bacteria Swarm Optimization (BSO)	Hybrid (Integrative)	Multi-stage	Socha
	Wahid and Hussin (2015)	CB-CTP	Harmony Search Algorithm (HSA)	Hybrid (Integrative)	Multi-stage	ITC2007-Track3
	Soria-Alcaraz et al. (2015)	PE-CTP	Parallel set of heuristics based on GA, Scatter Search, and discrete PSO	Hybrid (Collaborative)	Multi-stage	ITC2007-Track2
2016	Bellio et al. (2016)	CB-CTP	Simulated Annealing (SA)	Simulated Annealing	Single-stage	ITC2007-Track3
	Abdelhalim and El Khayat (2016)	UCTP	Utilization-base Genetic Algorithm	Evolutionary Algorithm	Single-stage	Real-World
	Kenekayoro and Zipamone (2016)	CB-CTP	Ant System	Swarm Intelligence	Single-stage	ITC2007-Track3
	Thepphakorn et al. (2016)	UCTP	Multiple Objective Cuckoo Search based Timetabling (MOCST)	Swarm Intelligence	Single-stage	Real-World
	Yousef et al. (2016)	CB-CTP	GPU Based Genetic Algorithm	Hybrid (Integrative)	Single-stage	ITC2007-Track3
	Lohpetch and Jaengchuea (2016)	PE-CTP	Hybrid NSGA-II with Two LS techniques and a TS heuristic (HNSGA2LTS)	Hybrid (Integrative)	Single-stage	MN
2017	Goh et al. (2017)	PE-CTP	Tabu Search with Sampling and Perturbation + Simulated Annealing with Reheating (TSSP + SAR)	Hybrid (Collaborative)	Multi-stage	Socha, ITC2002, ITC2007-Track2
	Kiefer et al. (2017)	CB-CTP	Adaptive Large Neighborhood Search (ALNS)	Large Neighborhood Search	Single-stage	ITC2007-Track3
	Song et al. (2017)	UCTP	Genetic Algorithm	Evolutionary Algorithm	Single-stage	Real-World
	Zhang et al. (2017)	UCTP	Ecogeography-Based Optimization (EBO)	Evolutionary Algorithm	Single-stage	Real-World
	Wahid (2017)	CB-CTP	Harmony Search Algorithm (HSA)	Hybrid (Integrative)	Multi-stage	Real-World
	Aziz et al. (2017)	UCTP	Honey-Bee Mating Optimization with Adaptive Guided Variable Neighborhood Search (HBMO-AGVNS)	Hybrid (Integrative)	Multi-stage	Socha
	Feng et al. (2017)	UCTP	Hybrid Genetic Algorithm (HGA) and Mixed Integer Linear Programming (MILP)	Hybrid (Integrative)	Single-stage	ITC2007
2018	Akkan and Gülcü (2018)	CB-CTP	Multi-Objective Genetic Algorithm + Hill Climbing + Simulated Annealing (MOGA + HC + SA)	Hybrid (Integrative)	Single-stage	ITC2007-Track3
	Song et al. (2018)	UCTP	Iterated Local Search (ILS)	Simulated Annealing	Multi-stage	Hard
	Gozali and Fujimura (2018)	UCTP	Reinforced asynchronous Island Model Genetic Algorithm (RIMGA)	Evolutionary Algorithm	Single-stage	Real-World
	Matias et al. (2018a)	UCTP	Genetic Algorithm with Guided Search Technique	Hybrid (Integrative)	Single-stage	Real-World
	Matias et al. (2018b)	CB-CTP	Genetic Algorithm with Guided Search and Self-Adaptive Neighborhood Strategies	Hybrid (Integrative)	Single-stage	ITC2007-Track3
2019	Goh et al. (2019b)	PE-CTP	Tabu Search with Sampling and Perturbation + Simulated Annealing with Improved Reheating and Learning (TSSP + SAIRL)	Hybrid (Collaborative)	Multi-stage	Socha, ITC2002, ITC2007-Track2
	Hossain et al. (2019)	UCTP	Particle Swarm Optimization with Selective Search (PSOSS)	Swarm Intelligence	Single-stage	Real-World
	Thepphakorn and Pongcharoen (2019)	UCTP	Particle Swarm Optimization Based Timetabling (PSOT)	Swarm Intelligence	Single-stage	Real-World
	Rezaeipanah et al. (2019)	UCTP	Parallel Genetic Algorithm and Local Search (PGALS)	Hybrid (Collaborative)	Single-stage	BenPaechter
2020	Chen et al. (2020)	UCTP	Tabu search algorithm with controlled randomization strategy (TSCR)	Tabu Search	Single-stage	Hard
	Goh et al. (2020)	PE-CTP	Tabu Search with Sampling and Perturbation with Iterated Local Search + Simulated Annealing with Reheating with Two Preliminary runs (TSSP-ILS + SAR-2P)	Simulated Annealing	Multi-stage	Hard, Socha, ITC2002, ITC2007-Track2
	Gülcü and Akkan (2020)	CB-CTP	Multi-Objective Simulated Annealing for Single Disruption (MOSA-SD), Multi-Objective Simulated Annealing for multiple disruptions with Sample Average Approximation (MOSA-SAA)	Simulated Annealing	Multi-stage	ITC2007-Track3

(continued on next page)

Table 12 (*continued*)

Year	Authors	Problem	Algorithm	Approach	Single/Multi-Stage	Dataset
	Gozali et al. (2020)	PE-CTP	Localized Island Model Genetic Algorithm with Dual Dynamic Migration Policy (DM-LIMGA)	Evolutionary Algorithm	Single-stage	ITC2007-Track2, Telkom University
	Thepphakorn and Pongcharoen (2020)	UCTP	Hybrid Self-adaptive Cuckoo Search-based Timetabling (HSCST)	Hybrid (Integrative)	Single-stage	Real-World
	Thepphakorn et al. (2020)	UCTP	Hybrid Particle Swarm Optimization-based Timetabling (HPSOT)	Hybrid (Integrative)	Single-stage	Real-World
	Unprasertporn and Lohpetch (2020)	PE-CTP	Hybrid Genetic-based Discrete Particle Swarm Optimization algorithm hybridizing with two different local search algorithms including Local Search and Tabu Search (HGDPSOLTS)	Hybrid (Integrative)	Multi-stage	MN
2021	Rezaeipanah et al. (2021)	PE-CTP	Improved Parallel Genetic Algorithm and Local Search (IPGALS)	Hybrid (Integrative)	Multi-stage	BenPaechter, ITC2007-Track2
	Thepphakorn et al. (2021)	UCTP	Hybrid Particle Swarm Optimization-based Timetabling (HPSOT)	Hybrid (Integrative)	Single-stage	Real-World
	Song et al. (2021)	CB-CTP	Competition-guided Multi-neighborhood Local Search (CMLS)	Hybrid (Integrative)	Multi-stage	ITC2007-Track3
2022	Akkan et al. (2022)	CB-CTP	Multi-Objective Simulated Annealing (MOSA)	Simulated Annealing	Single-stage	ITC2007-Track3
	Cruz-Rosales et al. (2022)	UCTP	Simulated Annealing with Cooperative Processes (SACP)	Simulated Annealing	Single-stage	MN
	Wong et al. (2022)	PE-CTP	Genetic Algorithm with Tabu Search with Sampling and Perturbation (GA + TSSP)	Hybrid (Integrative)	Multi-stage	Real-World

have achieved high performance on the UCTP benchmark datasets and appear to be the trend (Chen et al., 2021).

This paper surveys (hybrid) meta-heuristic approaches for solving the UCTP proposed since 2015. The approaches are reviewed, categorized, analyzed, and compared. Moreover, a detailed introduction of the UCTP problem and features of its benchmark datasets is provided. Finally, trends in the field are identified, and research opportunities in the UCTP are discussed. We strongly believe that this survey paper can be of great importance to the OR community in planning their research in the UCTP domain.

This survey paper reveals that there has been a shift towards hybrid meta-heuristic approaches in the literature since 2015 (refer to Table 9). These methods are common in both real-world and benchmark UCTP. In addition, a rise in incorporating mathematical optimization and matheuristics in the UCTP can be observed, especially on the ITC2019 benchmark dataset, as shown in Table 6.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

Abdelhalim, E. A., & El Khayat, G. A. (2016). A utilization-based genetic algorithm for solving the university timetabling problem (uga). *Alexandria Engineering Journal*, *55*, 1395–1409.

Akkan, C., & Gülcü, A. (2018). A bi-criteria hybrid genetic algorithm with robustness objective for the course timetabling problem. *Computers & Operations Research*, *90*, 22–32.

Akkan, C., Gülcü, A., & Kuş, Z. (2022). Bi-criteria simulated annealing for the curriculum-based course timetabling problem with robustness approximation. *Journal of Scheduling*, *25*, 477–501.

Aschinger, M., Applebee, S., Bucur, A., Edmonds, H., Hungerländer, P., & Maier, K. (2018). New constraints and features for the University Course Timetabling Problem. In *Operations research proceedings 2016* (pp. 95–101). Springer.

Atsuta, M., Nonobe, K., & Ibaraki, T. (2008). Itc2007 track 2, an approach using general csp solver. In *Practice and theory of automated timetabling (PATAT 2008)* (pp. 19–22).

Aziz, R. A., Ayob, M., Othman, Z., Ahmad, Z., & Sabar, N. R. (2017). An adaptive guided variable neighborhood search based on honey-bee mating optimization algorithm for the course timetabling problem. *Soft Computing*, *21*, 6755–6765.

Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, *86*, 43–59.

Babaei, H., Karimpour, J., & Hadidi, A. (2019). Generating an optimal timetabling for multi-departments common lecturers using hybrid fuzzy and clustering algorithms. *Soft Computing*, *23*, 4735–4747.

Badoni, R. P., & Gupta, D. K. (2015a). A hybrid algorithm for university course timetabling problem. *Innovative Systems Design and Engineering*, *6*, 60–66.

Badoni, R. P., & Gupta, D. K. (2015b). A new algorithm based on students groupings for university course timetabling problem. In *2015 2nd international conference on recent advances in engineering & computational sciences (RAECS)* (pp. 1–5). IEEE.

Bashab, A., Ibrahim, A. O., AbedElgabar, E. E., Ismail, M. A., Elsafi, A., Ahmed, A., & Abraham, A. (2020). A systematic mapping study on solving university timetabling problems using meta-heuristic algorithms. *Neural Computing & Applications*, *32*, 17397–17432.

Bellio, R., Ceschia, S., Di Gaspero, L., Schaerf, A., & Urli, T. (2016). Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem. *Computers & Operations Research*, *65*, 83–92.

Beni, G., & Wang, J. (1993). Swarm intelligence in cellular robotic systems. In *Robots and biological systems: Towards a new bionics?* (pp. 703–712). Springer.

Bettinelli, A., Cacchiani, V., Roberti, R., & Toth, P. (2015). An overview of curriculum-based course timetabling. *Top*, *23*, 313–349.

Bianchi, L., Dorigo, M., Gambardella, L. M., & Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, *8*, 239–287.

Bibai, J., Savéant, P., Schoenauer, M., & Vidal, V. (2010). On the generality of parameter tuning in evolutionary planning. In *Proceedings of the 12th annual conference on genetic and evolutionary computation* (pp. 241–248).

Blum, C., Puchinger, J., Raidl, G. R., & Roli, A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, *11*, 4135–4151.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, *35*, 268–308.

Blum, C., Roli, A., & Sampels, M. (2008). *Hybrid metaheuristics: An emerging approach to optimization: Vol. 114*. Springer.

Bonabeau, E., Theraulaz, G., & Dorigo, M. (1999). *Swarm intelligence*. Springer.

Burke, E. K., & Kendall, G. (2014). *Search methodologies: Introductory tutorials in optimization and decision support techniques*. Springer.

Bykov, Y. (2003). The description of the algorithm for international timetabling competition. In *International timetable competition*.

Cambazard, H., Hebrard, E., O'Sullivan, B., & Papadopoulos, A. (2007). Submission to ICT: Track 2. In *International Timetabling Competition*.

Černỳ, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, *45*, 41–51.

Chen, M., Tang, X., Song, T., Wu, C., Liu, S., & Peng, X. (2020). A tabu search algorithm with controlled randomization for constructing feasible university course timetables. *Computers & Operations Research*, *123*, Article 105007.

Chen, M. C., Goh, S. L., Sabar, N. R., Kendall, G., et al. (2021). A survey of University Course Timetabling Problem: Perspectives, trends and opportunities. *IEEE Access*.

Chiarandini, M., Fawcett, C., & Hoos, H. H. (2008). A multiphase modular heuristic solver for post enrollment course timetabling. *Journal of Scheduling*, 2–3.

Colorni, A., Dorigo, M., Maniezzo, V., et al. (1991). Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life* (pp. 134–142).

Cordeau, J. F., Jaumard, B., & Morales, R. (2003). Efficient timetabling solution with tabu search. In *International timetable competition*.

Cruz-Rosales, M. H., Cruz-Chávez, M. A., Alonso-Pecina, F., Peralta-Abarca, J. d. C., Ávila-Melgar, E. Y., Martínez-Bahena, B., & Enríquez-Urbano, J. (2022). Metaheuristic with cooperative processes for the university course timetabling problem. *Applied Sciences*, *12*, 542.

Delorme, X., Gandibleux, X., & Degoutin, F. (2010). Evolutionary, constructive and hybrid procedures for the bi-objective set packing problem. *European Journal of Operational Research*, *204*, 206–217.

Di Gaspero, L., McCollum, B., & Schaerf, A. (2007). The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3). Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1.0. Belfast, United Kingdom: Queen's University.

Du, K. L., Swamy, M. N. S., et al. (2016). *Search and optimization by metaheuristics. Techniques and algorithms inspired by nature*.

Duman, E., Uysal, M., & Alkaya, A. F. (2011). Migrating birds optimization: A new meta-heuristic approach and its application to the quadratic assignment problem. In *European conference on the applications of evolutionary computation* (pp. 254–263). Springer.

Eiben, A. E., Smith, J. E., et al. (2003). *Introduction to evolutionary computing: Vol. 53*. Springer.

Feng, X., Lee, Y., & Moon, I. (2017). An integer program and a hybrid genetic algorithm for the university timetabling problem. *Optimization Methods & Software*, *32*, 625–649.

Fong, C. W., Asmuni, H., & McCollum, B. (2015). A hybrid swarm-based approach to university timetabling. *IEEE Transactions on Evolutionary Computation*, *19*, 870–884.

Fraser, A. S. (1957). Simulation of genetic systems by automatic digital computers I. Introduction. *Australian Journal of Biological Sciences*, *10*, 484–491.

Gashi, E., & Sylejmani, K. (2019). Simulated annealing with penalization for university course timetabling.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation*, *76*, 60–68.

Ghasemi, E., Moradi, P., & Fathi, M. (2015). Integrating ABC with genetic grouping for university course timetabling problem. In *2015 5th international conference on computer and knowledge engineering (ICCKE)* (pp. 24–29). IEEE.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, *13*, 533–549.

Glover, F. (1989). Tabu search—part I. *ORSA Journal on Computing*, *1*, 190–206.

Glover, F. (1990). Tabu search—part II. *ORSA Journal on Computing*, *2*, 4–32.

Goh, S. L., Kendall, G., & Sabar, N. R. (2017). Improved local search approaches to solve the post enrolment course timetabling problem. *European Journal of Operational Research*, *261*, 17–29.

Goh, S. L., Kendall, G., & Sabar, N. R. (2019a). Monte Carlo tree search in finding feasible solutions for course timetabling problem. *International Journal on Advanced Science Engineering Information Technology*, *9*, 1936.

Goh, S. L., Kendall, G., & Sabar, N. R. (2019b). Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem. *Journal of the Operational Research Society*, *70*, 873–888.

Goh, S. L., Kendall, G., Sabar, N. R., & Abdullah, S. (2020). An effective hybrid local search approach for the post enrolment course timetabling problem. *Opsearch*, *57*, 1131–1163.

Goh, S. L., Sabar, N. R., Abdullah, S., Kendall, G., et al. (2022). A 2-stage approach for the nurse rostering problem. *IEEE Access*, *10*, 69591–69604.

Gozali, A. A., & Fujimura, S. (2018). Reinforced island model genetic algorithm to solve university course timetabling. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *16*, 2747–2755.

Gozali, A. A., Kurniawan, B., Weng, W., & Fujimura, S. (2020). Solving university course timetabling problem using localized island model genetic algorithm with dual dynamic migration policy. *IEEJ Transactions on Electrical and Electronic Engineering*, *15*, 389–400.

Gülcü, A., & Akkan, C. (2020). Robust university course timetabling problem subject to single and multiple disruptions. *European Journal of Operational Research*, *283*, 630–646.

Haddad, O. B., Afshar, A., & Marino, M. A. (2006). Honey-bees mating optimization (hbmo) algorithm: A new heuristic approach for water resources optimization. *Water Resources Management*, *20*, 661–680.

Herres, B., & Schmitz, H. (2021). Decomposition of university course timetabling. *Annals of Operations Research*, *302*, 405–423.

Holm, D. S., Mikkelsen, R. Ø., Sørensen, M., & Stidsen, T. R. (2019). A mip based approach for international timetabling competation 2019. In *Proceedings of the international timetabling competition 2020*.

Hossain, S. I., Akhand, M. A. H., Shuvo, M. I. R., Siddique, N., & Adeli, H. (2019). Optimization of university course scheduling problem using particle swarm optimization with selective search. *Expert Systems with Applications*, *127*, 9–24.

Hosseini Shirvani, M., & Noorian Talouki, R. (2022). Bi-objective scheduling algorithm for scientific workflows on cloud computing platform with makespan and monetary cost minimization approach. *Complex & Intelligent Systems*, *8*, 1085–1114.

Ilyas, R., & Iqbal, Z. (2015). Study of hybrid approaches used for university course timetable problem (UCTP). In *2015 IEEE 10th conference on industrial electronics and applications (ICIEA)* (pp. 696–701). IEEE.

Jaengchuea, S., & Lohpetch, D. (2015). 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. In *2015 7th international conference on information technology and electrical engineering (ICITEE)* (pp. 29–34). IEEE.

Kampke, E. H., Scheideger, L. M., Mauri, G. R., & Boeres, M. C. S. (2019). A network flow based construction for a grasp sa algorithm to solve the university timetabling problem. In *Computational science and its applications–ICCSA 2019: Proceedings of the 19th international conference, part III* (pp. 215–231). Springer.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical Report tr06. Erciyes University, Engineering Faculty, Computer Engineering Department.

Kasemset, C., & Irohara, T. (2019). University course timetabling problem considering day and time pattern. *International Journal of Operational Research*, *36*, 375–398.

Kenekayoro, P., & Zipamone, G. (2016). Greedy ants colony optimization strategy for solving the curriculum based university course timetabling problem. arXiv preprint, arXiv:1602.04933.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (pp. 1942–1948). IEEE.

Kesur, K. B. (2013). Multiobjective optimization of delay and stops in traffic signal networks. In *Metaheuristics in water, geotechnical and transport engineering* (pp. 385–416).

Kiefer, A., Hartl, R. F., & Schnell, A. (2017). Adaptive large neighborhood search for the curriculum-based course timetabling problem. *Annals of Operations Research*, *252*, 255–282.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680.

Kostuch, P. (2003). Timetabling competition-sa-based heuristic. In *International timetabling competition*.

Lemos, A., Monteiro, P. T., & Lynce, I. (2021). Introducing UniCorT: An iterative university course timetabling tool with MaxSAT. *Journal of Scheduling*, 1–20.

Lewis, R. (2008). A survey of metaheuristic-based techniques for university timetabling problems. *OR Spektrum*, *30*, 167–190.

Lewis, R., & Paechter, B. (2007). Finding feasible timetables using group-based operators. *IEEE Transactions on Evolutionary Computation*, *11*, 397–413.

Lewis, R., & Thompson, J. (2015). Analysing the effects of solution space connectivity with an effective metaheuristic for the course timetabling problem. *European Journal of Operational Research*, *240*, 637–648.

Lindahl, M., Mason, A. J., Stidsen, T., & Sørensen, M. (2018). A strategic view of university timetabling. *European Journal of Operational Research*, *266*, 35–45.

Lohpetch, D., & Jaengchuea, S. (2016). A hybrid multi-objective genetic algorithm with a new local search approach for solving the post enrolment based course timetabling problem. In *Recent advances in information and communication technology 2016* (pp. 195–206). Springer.

López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., & Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, *3*, 43–58.

Lü, Z., & Hao, J. K. (2010). Adaptive tabu search for course timetabling. *European Journal of Operational Research*, *200*, 235–244.

Matias, J. B., Fajardo, A. C., & Medina, R. M. (2018a). A fair course timetabling using genetic algorithm with guided search technique. In *2018 5th international conference on business and industrial research (ICBIR)* (pp. 77–82). IEEE.

Matias, J. B., Fajardo, A. C., & Medina, R. M. (2018b). Examining genetic algorithm with guided search and self-adaptive neighborhood strategies for curriculum-based course timetable problem. In *2018 fourth international conference on advances in computing, communication & automation (ICACCA)* (pp. 1–6). IEEE.

Müller, T. (2009). Itc2007 solver description: A hybrid approach. *Annals of Operations Research*, *172*, 429–446.

Müller, T., & Murray, K. (2010). Comprehensive approach to student sectioning. *Annals of Operations Research*, *181*, 249–269.

Müller, T., Rudová, H., Müllerová, Z., et al. (2018). University course timetabling and international timetabling competition 2019. In *Proceedings of the 12th international conference on the practice and theory of automated timetabling (PATAT-2018)* (pp. 5–31).

Nagata, Y. (2018). Random partial neighborhood search for the post-enrollment course timetabling problem. *Computers & Operations Research*, *90*, 84–96.

Ngoo, C. M., Goh, S. L., Sabar, N. R., Abdullah, S., Kendall, G., et al. (2022). A survey of the nurse rostering solution methodologies: The state-of-the-art and emerging trends. *IEEE Access*.

Noorian Talouki, R., Hosseini Shirvani, M., & Motameni, H. (2021). A hybrid meta-heuristic scheduler algorithm for optimization of workflow scheduling in cloud heterogeneous computing environment. *Journal of Engineering, Design and Technology*, *20*, 1581–1605.

NoorianTalouki, R., Shirvani, M. H., & Motameni, H. (2022). A heuristic-based task scheduling algorithm for scientific workflows in heterogeneous cloud computing platforms. *Journal of King Saud University: Computer and Information Sciences*, *34*, 4902–4913.

Osman, I. H., & Kelly, J. P. (1997). Meta-heuristics theory and applications. *Journal of the Operational Research Society*, *48*, 657.

Oude Vrielink, R. A., Jansen, E. A., Hans, E. W., & van Hillegersberg, J. (2019). Practices in timetabling in higher education institutions: a systematic review. *Annals of Operations Research*, *275*, 145–160.

Pandey, J., & Sharma, A. K. (2016). Survey on university timetabling problem. In *2016 3rd international conference on computing for sustainable global development (INDIACom)* (pp. 160–164). IEEE.

Rezaeipanah, A., Abshirini, Z., & Zade, M. B. (2019). Solving university course timetabling problem using parallel genetic algorithm. *International Journal of Scientific Research in Computer Science and Engineering*, *7*, 5–13.

Rezaeipanah, A., Matoori, S. S., & Ahmadi, G. (2021). A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search. *Applied Intelligence*, *51*, 467–492.

Rodríguez Maya, N., Flores, J. J., & Rodríguez, R. H. (2016). Performance comparison of evolutionary algorithms for university course timetabling problem. *Computación y Sistemas*, *20*, 623–634.

Rossi-Doria, O., Sampels, M., Birattari, M., Chiarandini, M., Dorigo, M., Gambardella, L. M., Knowles, J., Manfrin, M., Mastrolilli, M., Paechter, B., et al. (2002). A comparison of the performance of different metaheuristics on the timetabling problem. In *International conference on the practice and theory of automated timetabling* (pp. 329–351). Springer.

Rossi-Doria, O., Sampels, M., Birattari, M., Chiarandini, M., Dorigo, M., Gambardella, L. M., Knowles, J., Manfrin, M., Mastrolilli, M., Paechter, B., et al. (2003). A comparison of the performance of different metaheuristics on the timetabling problem. In *Practice and theory of automated timetabling IV: 4th international conference* (pp. 329–351). Springer (Selected Revised Papers 4).

Sabar, N. R., Goh, S. L., Turky, A., & Kendall, G. (2021). Population-based iterated local search approach for dynamic vehicle routing problems. *IEEE Transactions on Automation Science and Engineering*, *19*, 2933–2943.

Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review*, *13*, 87–127.

Shaker, K., Abdullah, S., & Alqudsi, A. (2015). Bacteria swarm optimisation approach for enrolment-based course timetabling problems. In *7th multidisciplinary international conference on scheduling. Theory and applications (MISTA 2015)* (pp. 515–525).

Shen, L. W., Asmuni, H., & Weng, F. C. (2015). A modified migrating bird optimization for university course timetabling problem. *Jurnal Teknologi*, *72*.

Shirvani, M. H. (2020). A hybrid meta-heuristic algorithm for scientific workflow scheduling in heterogeneous distributed computing systems. *Engineering Applications of Artificial Intelligence*, *90*, Article 103501.

Shirvani, M. H., & Talouki, R. N. (2021). A novel hybrid heuristic-based list scheduling algorithm in heterogeneous cloud computing environment for makespan optimization. *Parallel Computing*, *108*, Article 102828.

Silva, J., Varela, N., Varas, J., Lezama, O., Maco, J., & Villón, M. (2021). Comparison of bioinspired algorithms applied to the timetabling problem. In *Computational methods and data engineering* (pp. 427–437). Springer.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, *12*, 702–713.

Socha, K., Knowles, J., & Sampels, M. (2002). A max-min ant system for the university course timetabling problem. In *International workshop on ant algorithms* (pp. 1–13). Springer.

Song, K., Kim, S., Park, M., & Lee, H. S. (2017). Energy efficiency-based course timetabling for university buildings. *Energy*, *139*, 394–405.

Song, T., Chen, M., Xu, Y., Wang, D., Song, X., & Tang, X. (2021). Competition-guided multi-neighborhood local search algorithm for the university course timetabling problem. *Applied Soft Computing*, Article 107624.

Song, T., Liu, S., Tang, X., Peng, X., & Chen, M. (2018). An iterated local search algorithm for the University Course Timetabling Problem. *Applied Soft Computing*, *68*, 597–608.

Soria-Alcaraz, A. J., Carpio, M., Puga, H., Swan, J., Melin, P., Terashima, H., & Sotelo-Figueroa, A. M. (2015). Parallel meta-heuristic approaches to the course timetabling problem. In *Design of intelligent systems based on fuzzy logic, neural networks and nature-inspired optimization* (pp. 391–417). Springer.

Soria-Alcaraz, J. A., Özcan, E., Swan, J., Kendall, G., & Carpio, M. (2016). Iterated local search using an add and delete hyper-heuristic for university course timetabling. *Applied Soft Computing*, *40*, 581–593.

Tan, J. S., Goh, S. L., Kendall, G., & Sabar, N. R. (2021). A survey of the state-of-the-art of optimisation methodologies in school timetabling problems. *Expert Systems with Applications*, *165*, Article 113943.

Tanha, M., Hosseini Shirvani, M., & Rahmani, A. M. (2021). A hybrid meta-heuristic task scheduling algorithm based on genetic and thermodynamic simulated annealing algorithms in cloud computing environments. *Neural Computing & Applications*, *33*, 16951–16984.

Teoh, C. K., Wibowo, A., & Ngadiman, M. S. (2015). Review of state of the art for metaheuristic techniques in Academic Scheduling Problems. *Artificial Intelligence Review*, *44*, 1–21.

Thepphakorn, T., & Pongcharoen, P. (2019). Variants and parameters investigations of particle swarm optimisation for solving course timetabling problems. In *International conference on swarm intelligence* (pp. 177–187). Springer.

Thepphakorn, T., & Pongcharoen, P. (2020). Performance improvement strategies on Cuckoo Search algorithms for solving the university course timetabling problem. *Expert Systems with Applications*, *161*, Article 113732.

Thepphakorn, T., Pongcharoen, P., & Vitayasak, S. (2016). A new multiple objective cuckoo search for university course timetabling problem. In *International workshop on multi-disciplinary trends in artificial intelligence* (pp. 196–207). Springer.

Thepphakorn, T., Sooncharoen, S., & Pongcharoen, P. (2020). Academic operating costs optimisation using hybrid MCPSO based course timetabling tool. In *International conference on blended learning* (pp. 338–350). Springer.

Thepphakorn, T., Sooncharoen, S., & Pongcharoen, P. (2021). Particle swarm optimisation variants and its hybridisation ratios for generating cost-effective educational course timetables. *SN Computer Science*, *2*, 1–12.

Tindell, K. W., Burns, A., & Wellings, A. J. (1992). Allocating hard real-time tasks: An NP-hard problem made easy. *Real-Time Systems*, *4*, 145–165.

Unprasertporn, T., & Lohpetch, D. (2020). An outperforming hybrid discrete particle swarm optimization for solving the timetabling problem. In *2020 12th international conference on knowledge and smart technology (KST)* (pp. 18–23). IEEE.

Van Veldhuizen, D. A. (1999). *Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations*. Air Force Institute of Technology.

Wahid, J. (2017). Hybridizing harmony search with local search based metaheuristic for solving curriculum based university course timetabling.

Wahid, J., & Hussin, N. M. (2015). Solving curriculum based course timetabling by hybridizing local search based method within harmony search algorithm. In *International conference on soft computing in data science* (pp. 141–153). Springer.

Wahid, J., & Mohd Hussin, N. (2017). Hybrid harmony search with great deluge for UUM CAS curriculum based course timetabling. *Journal of Telecommunication, Electronic and Computer Engineering*, *9*, 33–38.

Wong, C. H., Goh, S. L., & Likoh, J. (2022). A genetic algorithm for the real-world university course timetabling problem. In *2022 IEEE 18th international colloquium on signal processing & applications (CSPA)* (pp. 46–50). IEEE.

Wren, A. (1995). Scheduling, timetabling and rostering—A special relationship? In *International conference on the practice and theory of automated timetabling* (pp. 46–75). Springer.

Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. In *2009 world congress on nature & biologically inspired computing (NaBIC)* (pp. 210–214). IEEE.

Yousef, A. H., Salama, C., Jad, M. Y., El-Gafy, T., Matar, M., & Habashi, S. S. (2016). A GPU based genetic algorithm solution for the timetabling problem. In *2016 11th international conference on computer engineering & systems (ICCES)* (pp. 103–109). IEEE.

Zamli, K. Z. (2018). Enhancing generality of meta-heuristic algorithms through adaptive selection and hybridization. In *2018 international conference on information and communications technology (ICOIACT)* (pp. 67–71). IEEE.

Zhang, M. X., Zhang, B., & Qian, N. (2017). University course timetabling using a new ecogeography-based optimization algorithm. *Natural Computing*, *16*, 61–74.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, *3*, 257–271.