IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 6 August 2024, accepted 13 September 2024, date of publication 19 September 2024, date of current version 11 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3463721

SURVEY

Timetabling Problems and the Effort Toward Generic Algorithms: A Comprehensive Survey

I. GUSTI AGUNG PREMANANDA, ARIS TJAHYANTO, AND AHMAD MUKLASON
Department of Information Systems, Institut Teknologi Sepuluh Nopember (ITS), Kampus ITS Sukolilo Surabaya, Surabaya, East Java 60111, Indonesia
Corresponding author: Ahmad Muklason (muklason@its.ac.id)

ABSTRACT The timetabling problem, a well-known NP-Hard optimization challenge, spans multiple domains such as education, healthcare, sports, and transportation. Due to its computational complexity, heuristic methods have become the dominant method for solving these problems. However, a lack of consistent comparisons across studies has led to difficulties in evaluating the effectiveness of these algorithms. This paper provides a comprehensive survey of 64 studies published between 2012 and 2022, focusing on timetabling algorithms and their performance on established benchmarks. The algorithms are categorized into metaheuristics, hybrid metaheuristics, and hyper-heuristics, with their efficacy evaluated across six major benchmarks. The analysis highlights the superiority of single-solution-based metaheuristics, with Simulated Annealing emerging as the most effective algorithm, particularly for educational timetabling problems. Additionally, the challenge of developing generic algorithms capable of performing across different timetabling problem domains is addressed. Despite advances in the field, cross-domain adaptability remains a critical area for further exploration. This survey serves as a guide for future research, providing insights into algorithmic strategies that can enhance the efficiency and generalizability of solutions for timetabling problems.

INDEX TERMS Timetabling, heuristic, generic algorithm, survey.

I. INTRODUCTION

The timetabling problem is a significant optimization challenge encountered across various domains, including education, healthcare, sports, and transportation [1], [2], [3]. Due to its complex, combinatorial nature, it is classified as an NP-Hard problem [3]. Solving this problem with exact algorithms is computationally intensive and resource-demanding. Consequently, heuristic methods are frequently employed as a practical alternative [4]. While these methods can yield satisfactory results, they do not guarantee optimal solutions [5]. This limitation has driven researchers to develop new algorithms aimed at outperforming previous studies.

Research on timetabling has grown considerably since 1963, leading to the publication of numerous survey papers that review the state of the art. In the last decade, six notable surveys have been published. In 2015, Babaei et al. [6] reviewed university course timetabling problems, categorizing algorithms into operational research, metaheuristics, and novel intelligent methods, and concluded that metaheuristics were the most time-efficient. In 2019, Oude Vrielink et al. [7] conducted a systematic review of educational timetabling, finding that heuristic methods outperformed traditional algorithms, though the industry still favored older methods due to user interface design. In 2020, Bashab et al. [8] focused on metaheuristic methods and identified a lack of exploration and real-world testing in university timetabling problems. In 2021, Tan et al. [3] surveyed school timetabling and reported promising results from single-based metaheuristic solutions. In the same year, Chen et al. [1] reviewed university timetabling algorithms, covering operational research, metaheuristics, and hybrid approaches, while highlighting frequently used benchmark datasets. Lastly, in 2022, Ceschia et al. [9] conducted a comprehensive review of educational timetabling, comparing state-of-the-art solutions across six benchmarks.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiwu Li.

Despite these extensive surveys, there remain gaps in the existing literature. One notable issue is the absence of a comprehensive ranking system among various studies. This is important because many studies claim superior algorithm performance by comparing their results with specific prior research. However, these comparisons are inconsistent across papers. For instance, both Pillay [10] and Burke and Bykov [11], published in the same year, developed algorithms using the Carter benchmark and claimed superiority over earlier work, though the prior studies they compared against differed. This inconsistency makes it difficult to assess the true performance of these algorithms and identify the most promising methods and approaches for future exploration.

Another underexplored topic is the development of generic algorithms—algorithms that can adapt to various problems without requiring modifications while still delivering effective solutions [12]. Previous survey papers have not thoroughly examined this topic. Several studies claim that their algorithms perform well across multiple problem types, but the comparisons used to support these claims are inconsistent across the literature. This lack of consistency makes it challenging to assess how much the development of generic algorithms has advanced the timetabling field.

To address these issues, this paper surveys the timetabling research with a focus on two main areas. First, it provides a comprehensive ranking of papers that utilize well-known benchmarks, identifying the approaches and algorithms that have demonstrated superior performance. Second, it examines the effectiveness of generic algorithms in solving timetabling problems by comparing their performance with that of other algorithms surveyed in this study. Our analysis covers research from 2012 to 2022, focusing on heuristic methods applied to six well-established benchmarks, along with studies that employ two of the newest benchmarks. Through this review, we offer insights into promising algorithms and assess the impact of generic algorithms on timetabling research.

II. TIMETABLING PROBLEM, HEURISTIC AND GENERIC ALGORITHM

Timetabling is a scheduling process that involves allocating multiple events to specific time slots while considering various constraints. The goal is to create a feasible schedule that satisfies all hard constraints and minimizes violations of soft constraints [3]. Among the different timetabling problems, educational timetabling has been the most extensively studied [7], making it the primary source of benchmarks in the field.

Given the NP-Hard classification of timetabling problems, heuristic methods play a crucial role in finding feasible and optimized solutions [7], [13]. Among these, metaheuristics are one of the most widely used approaches. Metaheuristics are high-level procedures designed to solve a variety of optimization problems [14]. These approaches are generally divided into two categories: population-based metaheuristics (metaheuristic (P)) and single solution-based metaheuristics (metaheuristic (S)). Population-based metaheuristics, such as Genetic Algorithms, evaluate multiple potential solutions simultaneously. In contrast, single solution-based metaheuristics, like Simulated Annealing, focus on optimizing individual solutions [3].

The strengths and limitations of metaheuristic algorithms have led to the development of hybrid approaches. These approaches aim to combine the best aspects of different metaheuristic algorithms to improve problem-solving efficiency [3]. A common strategy in hybrid approaches is to merge population-based metaheuristics with single solution-based metaheuristics (hybrid (P + S)), such as combining Genetic Algorithms with Simulated Annealing. Alternatively, some hybrid approaches involve combining multiple single solution-based metaheuristics (hybrid (S + S)), such as the combination of Iterated Local Search and Simulated Annealing.

Another widely adopted approach is the hyper-heuristic approach. Hyper-heuristics are high-level methodologies designed to generate effective combinations of low-level heuristics for solving specific problem instances or classes of problems [15]. The goal of hyper-heuristics is to enhance the generality of heuristic algorithms, allowing them to adapt to a variety of problems. Hyper-heuristics can be divided into two main categories: heuristic selection, which focuses on selecting among pre-existing low-level heuristics, and heuristic generation, which creates new heuristics from components of existing ones. These categories are further divided into perturbation approaches, which modify complete solutions, and construction approaches, which build new solutions [16]. As a result, hyper-heuristics can be classified into four combinations: hyper-heuristic with selection perturbation (S-P), hyper-heuristic with selection construction (S-C), hyper-heuristic with generation perturbation (G-P), and hyper-heuristic with generation construction (G-C).

In the development of algorithms for timetabling, the objective is not always to outperform previous studies. Another key goal is to develop generic algorithms capable of solving a wide range of timetabling problems. A generic algorithm should be adaptable to various problems without the need for modifications while still producing effective solutions [12]. In the context of timetabling, three levels of generality have been identified: algorithms applicable to problems within the same benchmark set, those applicable to different benchmark sets, and those capable of solving various types of timetabling problems [17].

III. OVERVIEW OF SELECTED LITERATURE

To conduct a comparative analysis, we selected studies based on a set of criteria. First, we chose studies published in journals between 2012 and 2022. To ensure quality, we limited our selection to journals ranked at least Q2. Second, we focused on studies that used benchmarks to test their algorithms, specifically selecting benchmarks that were widely used across multiple studies and offered a variety of constraints. We identified six benchmarks that met

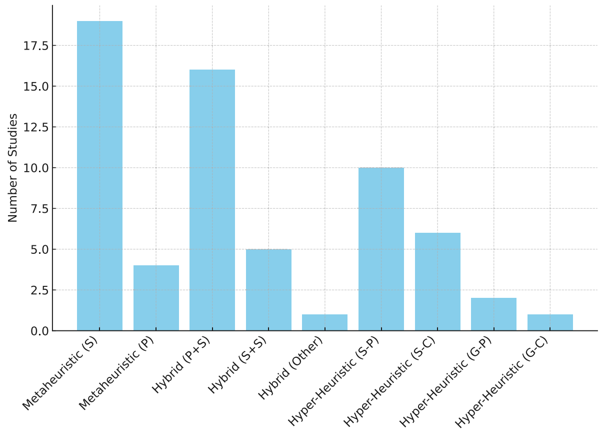

FIGURE 1. Distribution approach.

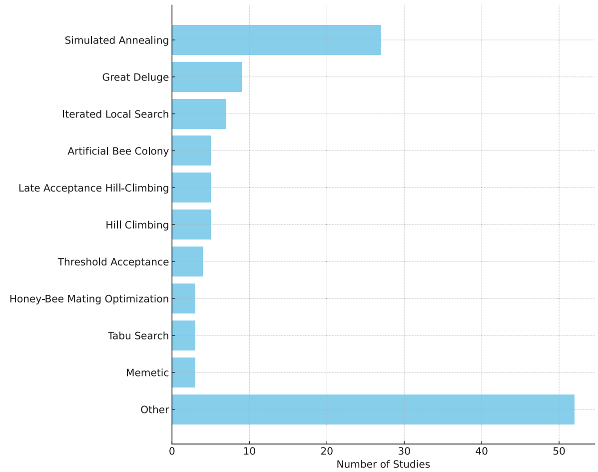

FIGURE 2. Distribution algorithms.

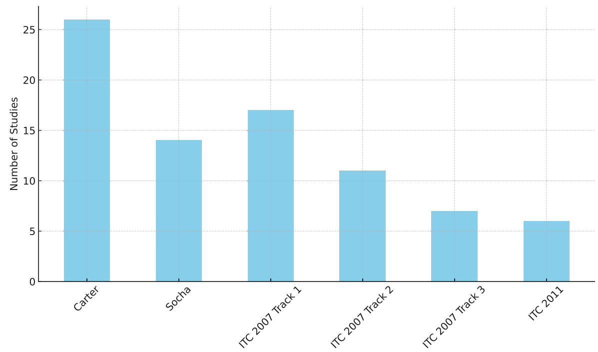

FIGURE 3. Distribution benchmark.

these criteria: the Carter, Socha, International Timetabling Problem (ITC) 2007 (across three different tracks), and ITC 2011 benchmarks. As a result, 61 studies were selected for our comparative analysis. Additionally, we included three more studies—two that used the ITC 2019 benchmark and one that employed the 2021 benchmark—to examine the outcomes of heuristic methods in the most recent benchmarks.

Among the 64 articles selected, the distribution of approaches was fairly balanced: 23 articles used a metaheuristic approach, 22 adopted a hybrid approach, and 19 implemented a hyper-heuristic approach. In the metaheuristic category, metaheuristic (S) was predominant, appearing in 19 articles, while metaheuristic (P) was used in only four articles. For hybrid approaches, hybrid (P + S) was dominant used in 16 articles. Five articles utilized a hybrid (S + S), and one article employed a unique hybrid approach combining a single solution with a construction method (hybrid(other)). In the hyper-heuristic category, selection-based approaches were the most popular, with ten articles using hyper-heuristics (S-P) and six using hyper-heuristic (S-C). In contrast, only three articles used generation-based approaches, with two employing hyper-heuristic (G-P) and one using hyper-heuristic (G-C). Figure 1 shows a detailed breakdown of the distribution of approaches.

In terms of algorithms, 49 distinct algorithms were developed across the 64 studies. Single solution-based algorithms were the most common, with Simulated Annealing being the most frequently used, appearing in 27 articles. Other widely used algorithms include the Great Deluge, used in 9 articles, Iterated Local Search, featured in 7 studies, and Hill Climbing and Late Acceptance Hill Climbing, each appearing in 5 articles. Population-based approaches were less common, with the Artificial Bee Colony algorithm appearing in five articles, while Memetic and Honey Bee Mating Optimization algorithms were each used in three articles. Figure 2 illustrates the distribution of algorithms used in the reviewed studies.

Regarding benchmarks, the ITC 2007 benchmark was the most commonly used, appearing in 35 articles. This benchmark is divided into three tracks: Track 1 (exam timetabling), Track 2 (post-enrollment course timetabling), and Track 3 (curriculum course timetabling). The second most popular benchmark was the Carter benchmark, which was used in 26 articles. The Socha and ITC 2011 benchmarks were less frequently used, appearing in 14 and 6 articles, respectively. Figure 3 provides a detailed visualization of the benchmark usage frequency.

IV. COMPARATIVE ANALYSIS OF LITERATURE FINDINGS

This section presents the comparison results from the 64 studies discussed in Section III. We evaluate the outcomes across six benchmarks by considering the best results from each study. The 'best result' was chosen as the comparison metric, as not all studies report the average results of their findings. The comparison ranks each study within its respective dataset, and the average rank across all datasets is used as the final outcome. Studies that do not report results for certain datasets are excluded from the rankings for those specific datasets. Subsections A to F provide a detailed breakdown of the comparison results and the corresponding findings. In subsection G, we also discuss the heuristic methods used to solve the new benchmarks (ITC 2019 and ITC 2021).

A. CARTER BENCHMARK

The Carter benchmark, also known as the Toronto benchmark, was introduced by Carter in 1996 [18]. It addresses the uncapacitated examination timetabling problem across multiple universities by eliminating all side constraints. Compared to five other benchmarks, the Carter benchmark has the fewest constraints, with a hard constraint ensuring that no student is scheduled for multiple exams simultaneously. It also includes a soft constraint designed to spread conflicting exams apart, giving students more time to prepare [19]. This benchmark consists of 18 datasets and imposes no running time limit.

In this benchmark, 26 studies were reviewed. Most of these studies tested their algorithms on only 12 datasets, and the comparison results are based on these datasets. Table 1 presents the rankings of the studies. Among the metaheuristic (S) approaches, the study by [20] using Simulated Annealing and the study by [11] introducing the Great Deluge algorithm achieved notable success, ranking first and third, respectively. In the hybrid category, the work by [21], which integrated Cellular Memetics with Threshold Acceptance, secured second place. Within the hyper-heuristic (P-S) category, the study by [22], also employing Simulated Annealing, ranked fourth. However, not all studies using these approaches consistently achieved high rankings. For instance, the studies by [23], [24], and [25] ranked 15th, 17th, and 19th, despite using similar metaheuristic (S) approaches.

Other approaches did not perform as well as the metaheuristic (S) and hybrid approaches in this benchmark. Both metaheuristics (P) and hyper-heuristics (G-P) yielded only average outcomes. More notably, hyper-heuristics (S-C) placed three studies in the lowest three positions, with two additional studies ranking 16th and 18th. A single study using hyper-heuristics (G-C) [26] ranked 23rd, indicating that the construction-based strategy in this benchmark tends to underperform.

Figure 4 presents a scatter plot that illustrates the distribution of different approaches. Each study is represented by distinct symbols and colors corresponding to its approach category. The y-axis lists the research studies, while the x-axis represents their rankings. The figure shows that some studies using metaheuristic (S), hybrid (P + S), and hyper-heuristic (P-S) approaches achieve superior performance compared to others. However, some studies employing the same approaches show average or below-average results, likely due to differences in algorithmic design and parameter tuning. Despite this variability, the findings suggest that these approaches are promising for addressing timetabling problems.

The Carter benchmark, with its minimal number of hard constraints, makes it easier for population-based algorithms to perform their search processes. However, the results indicate that studies using only population-based algorithms performed only slightly above average. This suggests that population-based algorithms may not be the most effective approach for solving timetabling problems.

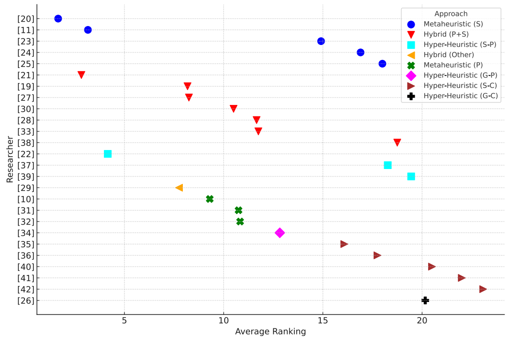

FIGURE 4. Distribution of average ranking in carter.

From an algorithmic perspective, Simulated Annealing and the Great Deluge showed strong performance in this benchmark. These two algorithms consistently ranked above average in the studies that employed them. The Simulated Annealing technique, applied by [20], [22], [27], and [28], achieved ranks of 1, 4, 7, and 12, respectively. Similarly, the Great Deluge algorithm, used by [11], [19], and [29], secured ranks of 3, 5, and 6.

B. SOCHA BENCHMARK

The Socha benchmark, introduced by Socha in 2003 [43], was developed using a generator created by Ben Paechter. This benchmark addresses the post-enrollment course timetabling problem and presents a more complex set of hard and soft constraints compared to the Carter benchmark. The hard constraints include the following:

- No student should be scheduled for two or more events simultaneously.
- Each event must be allocated to a room that satisfies its specific requirements.
- A room can only host one event per time slot.

The Socha benchmark also includes three soft constraints aimed at optimizing the event distribution for students [44], [45]:

- Students should not have an event scheduled in the last time slot of the day.
- Students should not have more than two consecutive events.
- A student's schedule should not consist of just one event in a single day.

This benchmark consists of 12 datasets, divided into five small, five medium, and two large datasets. Like the Carter benchmark, no time limit is imposed for solving the Socha benchmark.

Fourteen studies that used the Socha benchmark were surveyed. All studies found optimal results on the small datasets, so these were excluded from the comparison. The large dataset number 2 was used in only one study, leading to its exclusion as well. Consequently, Table 2 presents the ranking of the studies based on the remaining datasets.

Metaheuristic (S) approaches produced superior results, securing the top four positions. Notably, three of the top four

TABLE 1. Comparison in carter benchmark.

No	Researcher	Approach	Algorithm	Average Ranking
1	[20]	Metaheuristic (S)	Simulated Annealing	1,66
2	[21]	Hybrid (P+S)	Cellular Memetic + Threshold Acceptance	2,83
3	[11]	Metaheuristic (S)	Great Deluge	3,16
4	[22]	Hyper-Heuristic (S-P)	Simulated Annealing	4,16
5	[29]	Hybrid (Other)	Great Deluge + Constructive Method	7,75
6	[19]	Hybrid (P+S)	Artificial Bee Colony + Great Deluge	8,18
7	[27]	Hybrid (P+S)	Bee Colony Optimization + Simulated Annealing + Late Acceptance Hil-Climbing	8,25
8	[10]	Metaheuristic (P)	Developmental Approach	9,3
9	[30]	Hybrid (P+S)	Artificial Bee Colony + Late Acceptance Hill-Climbing	10,5
10	[31]	Metaheuristic (P)	Harmony Search	10,75
11	[32]	Metaheuristic (P)	Honey-Bee Mating Optimization	10,83
12	[28]	Hybrid (P+S)	Bee Algorithm + Simulated Annealing + Late Acceptance Hill Climbing	11,66
13	[33]	Hybrid (P+S)	Intelligent Water Drops + Local Search Optimizer	11,75
14	[34]	Hyper-Heuristic (G-P)	Backus Naur Form Grammar + Genetic Algorithm	12,83
15	[23]	Metaheuristic (S)	B-hill Climbing	14,91
16	[35]	Hyper-Heuristic (S-C)	Estimation Distribution	16,08
17	[24]	Metaheuristic (S)	Tabu Search	16,9
18	[36]	Hyper-Heuristic (S-C)	Evolutionary Algorithm	17,75
19	[25]	Metaheuristic (S)	Ruin and Recreate	18
20	[37]	Hyper-Heuristic (S-P)	Memetic Algorithm	18,27
21	[38]	Hybrid (P+S)	Discrete Particle Swarm Optimization + Hill Climbing	18,75
22	[39]	Hyper-Heuristic (S-P)	Adaptive Selection Heuristic	19,45
23	[26]	Hyper-Heuristic (G-C)	Arithmetic and Hierarchical Heuristics	20,16
24	[40]	Hyper-Heuristic (S-C)	Evolutionary Multitasking Optimization	20,5
25	[41]	Hyper-Heuristic (S-C)	Memetic Algorithm + Hill Climbing	22
26	[42]	Hyper-Heuristic (S-C)	Graph Coloring	23,08

studies were authored by the same researcher: [44], [46], [47], who refined and modified Simulated Annealing techniques. The fourth study, by [45], introduced the Random Partial Neighbourhood Search algorithm.

In contrast, other approaches did not perform as well as the metaheuristic (S) approaches. Hybrid (P + S) approaches were the most commonly used among the studies but generally delivered unsatisfactory performance. Although one hybrid study managed to rank fifth, there was still a significant performance gap between it and the metaheuristic (S) approaches in terms of average ranking. Most hybrid studies ranked poorly, with their average rank exceeding nine. The only study that used a metaheuristic (P) approach, conducted by [32], ranked 11th. Another study by [48], which utilized a hyper-heuristic (S-P) approach, yielded mediocre results, ranking sixth.

Figure 5 presents the distribution of approaches on a scatter plot. This figure clearly demonstrates that metaheuristic (S) approaches outperform other approaches on the Socha benchmark. These results reinforce the conclusion that single solution-based approaches, such as metaheuristic (S), are better suited for solving timetabling problems than population-based approaches.

Regarding algorithmic performance, Simulated Annealing once again proved to be highly effective, achieving ranks of 1, 2, 3, and 6. However, the Great Deluge algorithm, which performed well on the Carter benchmark, did not achieve similar success on the Socha benchmark, ranking 8th and 14th. No other algorithm displayed a distinct advantage, as no single algorithm consistently delivered superior results across the various studies.

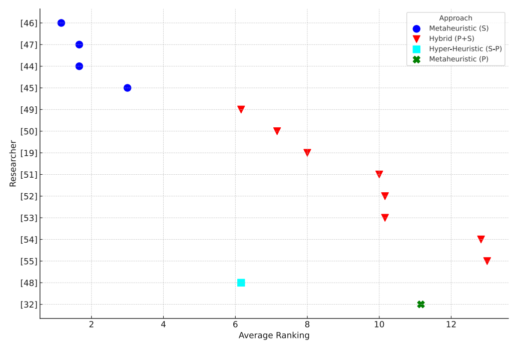

FIGURE 5. Distribution of average ranking in Socha.

C. ITC 2007 TRACK 1 BENCHMARK

The ITC 2007 Track 1 focuses on exam timetabling, presenting a higher level of complexity by introducing additional hard and soft constraints compared to the Carter and Socha benchmarks. These constraints are derived from real-world challenges [56]. The problem involves five hard constraints, which are:

- No student should be scheduled to take more than one examination at the same time.
- Room capacities should not be exceeded during any examination session.
- Period duration restrictions must be adhered to.

TABLE 2. Comparison in Socha benchmark.

No	Researcher	Approach	Algorithm	Average Ranking
1	[46]	Metaheuristic (S)	Simulated Annealing	1,16
2	[47]	Metaheuristic (S)	Simulated Annealing	1,66
3	[44]	Metaheuristic (S)	Simulated Annealing	1,66
4	[45]	Metaheuristic (S)	Random Partial Neighbourhood Search	3,00
5	[49]	Hybrid (P+S)	Honey-Bee Mating Optimization + Adaptive Guided Variable Neighbourhood Search	6,16
6	[48]	Hyper-Heuristic (S-P)	Simulated Annealing	6,16
7	[50]	Hybrid (P+S)	Gravitational Emulation + MPCA-ARDA	7,16
8	[19]	Hybrid (P+S)	Artificial Bee Colony + Great Deluge	8
9	[51]	Hybrid (P+S)	Harmony Search + Hill Climbing	10
10	[52]	Hybrid (P+S)	Artificial Bee Colony + Hill Climbing Optimizer	10,16
11	[53]	Hybrid (P+S)	Scatter Search + Hill Climbing + Iterated Local Search	10,16
12	[32]	Metaheuristic (P)	Honey-Bee Mating Optimization	11,16
13	[54]	Hybrid (P+S)	Genetic Algorithms + Local Search	12,83
14	[55]	Hybrid (P+S)	Electromagnetism-like Mechanism + Great Deluge	13

- Period-related hard constraints must be respected.
- Room-related hard constraints must be followed.

Additionally, the benchmark introduces seven soft constraints aimed at optimization:

- Students must sit for two consecutive exams.
- Students must sit for two exams in a single day.
- Exams for students are not spread out over a sufficient number of periods.
- Examinations of varying durations are scheduled within the same period.
- Examinations for larger classes are scheduled later in the exam session.
- Period-related soft constraints.
- Room-related soft constraints.

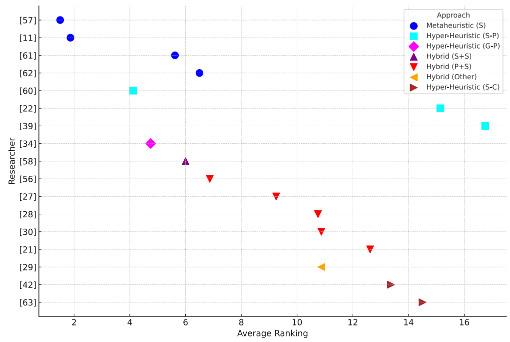

FIGURE 6. Distribution of average ranking in ITC 2007 track 1.

The ITC 2007 benchmark consists of 12 datasets and imposes a running time limit calculated by official software based on the specifications of the device being used.

This survey identified 19 studies that utilized this benchmark. However, two studies did not apply the time limit rule and were therefore excluded from the analysis. Among the remaining 17 studies, the majority utilized only the first eight datasets for testing. Therefore, our analysis concentrates on these eight datasets. Table 3 presents the comparison results for the 17 studies.

The findings indicate that metaheuristic (S) approaches outperformed other approaches. The top two studies using this approach were [57], which employed Step Counting Hill Climbing, and [11], which used the Great Deluge algorithm. Two other studies employing metaheuristic (S) approaches achieved respectable rankings, securing 5th and 7th positions. In terms of hybrid approaches, the only study by [58] that combined two single-solution approaches—Steepest Descent and Simulated Annealing—performed decently, securing the 6th rank. Conversely, hybrid approaches that combined population and single-solution algorithms generally performed poorly, ranking below the median. Another hybrid approach by [29] also failed to perform well, finishing 11th.

Among hyper-heuristic approaches, the study by [34], which employed a hyper-heuristic (G-P) approach combining Backus Naur Form Grammar with a Genetic Algorithm, secured a strong 3rd place. Another hyper-heuristic (S-P) study by [59], using Gene Expression Programming, performed well, ranking 4th. However, two other studies using the same approach by [22] and [39] were less successful, ranking 16th and 17th, respectively. Hyper-heuristic (S-C) approaches also underperformed, securing 14th and 15th positions.

Figure 6 presents a scatter plot showing the distribution of the different approaches. Once again, the metaheuristic (S) approach demonstrated superior performance, consistent with the results from the Carter and Socha benchmarks. While other approaches, such as hyper-heuristic (S-P) and hyper-heuristic (G-P), performed well, they lacked consistent results across all benchmarks. Hybrid approaches, particularly hybrid (P + S), underperformed, as seen with the Socha benchmark. Interestingly, the hybrid (S + S) approach, which combines single solution-based approaches, delivered better results than all hybrid (P + S) studies, further supporting the idea that single solution-based algorithms are better suited for timetabling problems.

In terms of algorithmic performance, a wide range of algorithms were employed in this benchmark. While Simulated Annealing dominated the previous two benchmarks, its performance here was less remarkable, with rankings of 5th,

TABLE 3. Comparison in ITC 2007 track 1 benchmark.

No	Researcher	Approach	Algorithm	Average Ranking
1	[57]	Metaheuristic (S)	Step Counting Hill Climbing	1,5
2	[11]	Metaheuristic (S)	Great Deluge	1,87
3	[60]	Hyper-Heuristic (S-P)	Gene Expression Programming	4,12
4	[34]	Hyper-Heuristic (G-P)	Backus Naur Form Grammar + Genetic Algorithm	4,75
5	[61]	Metaheuristic (S)	Simulated Annealing	5,62
6	[58]	Hybrid (S+S)	Steepest Descent + Simulated Annealing	6
7	[62]	Metaheuristic (S)	Simulated Annealing	6,5
8	[56]	Hybrid (P+S)	Memetic + Tabu Search	6,87
9	[27]	Hybrid (P+S)	Bee Colony Optimization + Simulated Annealing + Late Acceptance Hill-Climbing	9,25
10	[28]	Hybrid (P+S)	Bee Algorithm + Late Acceptance Hill Climbing + Simulated Annealing	10,75
11	[29]	Hybrid (other)	Great Deluge + Constructive Method	10,87
12	[30]	Hybrid (P+S)	Artificial Bee Colony + Late Acceptance Hill-Climbing	10,87
13	[21]	Hybrid (P+S)	Cellular Memetic + Threshold Acceptance	12,62
14	[42]	Hyper-Heuristic (S-C)	Graph Coloring	13,37
15	[63]	Hyper-Heuristic (S-C)	Iterative Adaptive	14,5
16	[22]	Hyper-Heuristic (S-P)	Simulated Annealing	15,14
17	[39]	Hyper-Heuristic (S-P)	Adaptive Selection Heuristic	16,75

6th, and 13th. The Great Deluge algorithm performed well by securing the 2nd position, the same rank it achieved in the Carter benchmark. A new algorithm, Step Counting Hill Climbing, introduced by [57], showed promise by achieving the top rank in this benchmark.

D. ITC 2007 TRACK 2 BENCHMARK

The ITC 2007 Track 2 benchmark is a competition benchmark designed for the post-enrollment course timetabling problem [64]. It shares the same soft constraints as the Socha benchmark. In terms of hard constraints, this benchmark also uses the same constraints as the Socha benchmark, with two additional constraints:

- Events must be assigned only to time slots that are predefined as "available" for those events.
- Where specified, events must be scheduled in the correct order within the week.

This benchmark comprises 24 datasets and imposes a runtime limit, following the same rules as ITC 2007 Track 1.

This paper reviewed twelve studies using this benchmark. One study that did not comply with the time limit rule was excluded from the comparison. Although most studies utilized all available datasets, the study by [65] limited its analysis to only the first 16 datasets. Therefore, the comparison for this study is based solely on those 16 datasets. The comparison results are presented in Table 4.

Metaheuristic (S) approaches delivered strong results, with five studies ranking in the top five. Notably, three of these top-ranking studies were authored by the same researcher, who employed similar algorithms. However, one study by [66] received the lowest rank, possibly due to its unique approach, which implemented a time-dependent mechanism that divided phases based on time, contrasting with the more commonly used metaheuristic (S) strategies. Other approaches, such as hybrid (P + S), hyper-heuristic (G-S), and hyper-heuristic (P-S), only achieved average or subpar results.

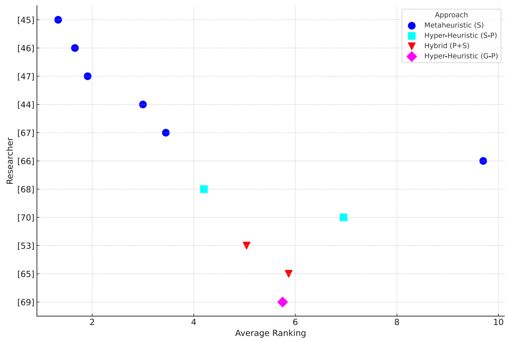

FIGURE 7. Distribution of average ranking in ITC 2007 track 2.

Figure 7 presents a scatter plot showing the distribution of approaches. The figure clearly demonstrates that the metaheuristic (S) approach consistently outperformed other approaches, aligning with the findings from the previous three benchmarks. This trend suggests that single-solution-based approaches are more effective for timetabling problems. In contrast, the other three approaches showed no positive trend, achieving mostly average results.

In terms of algorithms, Simulated Annealing continued to perform well, ranking between 2nd and 5th, with the exception of one study where it ranked last. This lower performance might be attributed to the use of the time-dependent mechanism discussed earlier. Random Partial Neighborhood Search also performed well, as seen in the Socha benchmark. Iterated Local Search, although popular and appearing in four studies, produced only average results.

E. ITC 2007 CURRICULUM-BASED COURSE TIMETABLING BENCHMARK

The ITC 2007 curriculum-based course timetabling represents the third problem introduced in the ITC 2007 competition. The dataset originates from a real-world case study at the University of Udine [71]. This problem involves four hard

TABLE 4. Comparison in ITC 2007 track 2 benchmark.

No	Researcher	Approach	Algorithm	Average Ranking
1	[45]	Metaheuristic (S)	Random Partial Neighbourhood Search	1,33
2	[46]	Metaheuristic (S)	Simulated Annealing	1,66
3	[47]	Metaheuristic (S)	Simulated Annealing	1,91
4	[44]	Metaheuristic (S)	Simulated Annealing	3
5	[67]	Metaheuristic (S)	Simulated Annealing	3,45
6	[68]	Hyper-Heuristic (S-P)	Iterated Local Search	4,2
7	[53]	Hybrid (P+S)	Scatter Search + Hill Climbing + Iterated Local Search	5,04
8	[69]	Hyper-Heuristic (G-P)	Iterated Local Search	5,75
9	[65]	Hybrid (P+S)	Ant Colony Optimization + Simulated Annealing	5,87
10	[70]	Hyper-Heuristic (S-P)	Iterated Local Search	6,95
11	[66]	Metaheuristic (S)	Simulated Annealing	9,7

constraints and four soft constraints [69]. The hard constraints are as follows:

- All lectures must be scheduled, each assigned to distinct periods.
- Two lectures cannot be scheduled in the same room at the same period.
- Lectures from courses within the same curriculum must be scheduled in different periods.
- Lectures must align with the availability and requirements of the instructors.

The four soft constraints are:

- For each lecture, the number of students attending must not exceed the room's capacity.
- Lectures for each course must be spread over the minimum required number of days.
- Lectures within the same curriculum should be scheduled consecutively (in adjacent periods).
- All lectures for a course should be scheduled in the same room.

This benchmark includes 21 datasets and, like ITC 2007 Track 1 and Track 2, follows the same time limit rule.

Of the nine studies reviewed on this benchmark, two did not adhere to the time limit rule and were excluded from the comparison. Most studies used all datasets except the study by [72], which only utilized the first 14 datasets. The comparison results are presented in Table 5.

In this benchmark, the Hyper-Heuristic (G-P) approach used by [69] outperformed other approaches, achieving the top rank across all datasets. Among hybrid approaches, the study by [56], which employed a hybrid (P + S) strategy, delivered strong results, securing 2nd place. However, another study using a similar approach by [55] only achieved 6th rank. Hybrid (S + S) approaches produced median results, securing 4th and 5th ranks. The metaheuristic (S) approach, while effective in previous benchmarks, was less dominant here, finishing 3rd and 7th.

The distribution of approaches is illustrated in Figure 8. While no single approach demonstrated overwhelming dominance in this benchmark, the results still reflect trends observed in previous benchmarks. The top-ranked Hyper-Heuristic (G-P) approach used a single solution-based strategy, reinforcing the efficacy of such approaches for

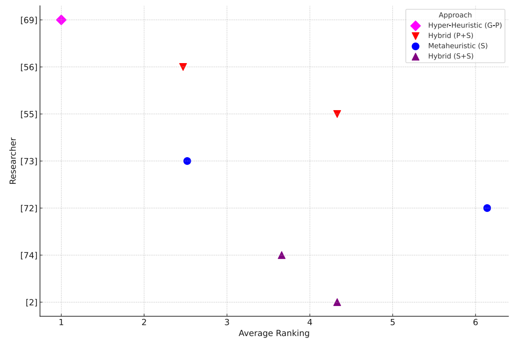

FIGURE 8. Distribution of average ranking in ITC 2007 track 3.

timetabling problems. The metaheuristic (S) approach also performed well, ranking 3rd, which continues to support the conclusion that single solution-based algorithms are well-suited for timetabling tasks.

In terms of algorithm performance, Simulated Annealing and Great Deluge, which showed strong results in previous benchmarks, were less successful in this benchmark. Simulated Annealing achieved 4th and 5th ranks, while Great Deluge placed 6th. Iterated Local Search, which had moderate success in ITC 2007 Track 2, delivered impressive results in this benchmark. Tabu Search also performed well, securing 2nd and 3rd ranks.

F. ITC 2011 BENCHMARK

The ITC 2011 competition focuses on high school timetabling problems, featuring real-world datasets sourced from schools in ten countries. This benchmark introduces a significantly larger set of constraints compared to the five previous benchmarks discussed. It includes 15 types of constraints, categorized as either hard or soft. A detailed description of these constraints can be found in [75].

We identified eight studies that utilized this benchmark and met our survey criteria. The benchmark consists of rounds, some with a time limit and others without. Of the eight studies, one reported results without a time limit, five presented results solely from rounds with time limits, and two provided results for both. To ensure a fair comparison, we only considered studies that provided results from rounds with time limits.

TABLE 5. Comparison in ITC 2007 track 3 benchmark.

No	Researcher	Approach	Algorithm	Average Ranking
1	[69]	Hyper-Heuristic (G-P)	Iterated Local Search	1
2	[56]	Hybrid (P+S)	Memetic + Tabu Search	2,47
3	[73]	Metaheuristic (S)	Adaptive Large Neighbourhood Search	2,52
4	[74]	Hybrid (S+S)	Simulated Annealing + Dynamic Tabu Search	3,66
5	[2]	Hybrid (S+S)	Simulated Annealing	4,33
6	[55]	Hybrid (P+S)	Electromagnetism-like Mechanism + Great Deluge	4,33
7	[72]	Metaheuristic (S)	Threshold Acceptance	6,14

Our comparison focused on the optimization of soft constraints, ensuring that all hard constraints were met to guarantee feasible solutions. To streamline the process, we excluded datasets for which no study produced feasible solutions. Consequently, we analyzed 12 datasets (BrazilInstance2, BrazilInstance3, BrazilInstance6, FinlandElementarySchool, FinlandSecondarySchool2, Aigio1stHighSchool10-11, ItalyInstance4, Kottenpark2003, Spanish school, WesternGreeceUniversity3, WesternGreeceUniversity4, and WesternGreeceUniversity5) from the original 18 datasets. Two studies by [76] and [77] failed to find feasible solutions for two datasets (BrazilInstance2 and BrazilInstance6), so these datasets were excluded when calculating ranks for these studies.

Table 6 presents the rankings of the six studies. We found that only three approaches were explored: metaheuristic (S), hybrid (S + S), and hyper-heuristic (S-P). The hybrid (S + S) approach delivered strong results, achieving ranks 1 and 3. Metaheuristic (S) also performed well, securing 2nd place. In contrast, hyper-heuristics underperformed, occupying the last three positions and displaying a noticeable gap in average ranking.

Figure 9 shows the distribution of approaches in a scatter plot. All studies employed single solution-based approaches using three different strategies. This reinforces the notion that single-solution algorithms are well-suited for timetabling problems, particularly for benchmarks of this nature, where the increasing number of constraints makes it more challenging for population-based algorithms to effectively optimize the solutions.

Regarding algorithms, Simulated Annealing performed well in two studies by [76] and [78], but yielded less favorable results in the study by [79]. Other algorithms, such as Iterated Local Search, Variable Neighborhood Search, and Late Acceptance Hill Climbing, also produced promising results.

G. ITC 2019 AND ITC 2021 BENCHMARK

The ITC 2019 benchmark focuses on course timetabling problems originating from real-world data. This benchmark comprises 30 distinct datasets, each containing 19 types of constraints, which are categorized as either hard or soft and vary across instances. Detailed descriptions of these constraints can be found in [82]. The main challenge of this benchmark lies not only in finding a feasible solution but

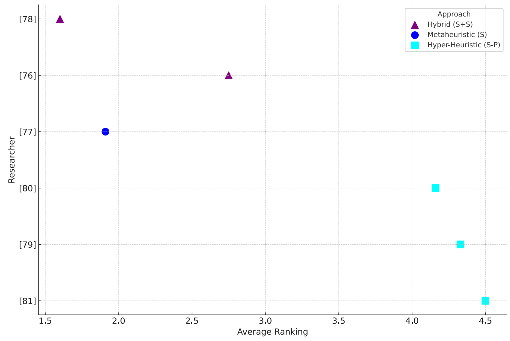

FIGURE 9. Distribution of average ranking in ITC 2011.

also in optimizing the solution. Consequently, only a limited number of studies have reported success with this benchmark. Among these, two studies employed heuristic approaches. The first, by [83], used a metaheuristic (S) approach based on Simulated Annealing and ranked 3rd out of 5 studies [84]. The second study, by [85], applied a metaheuristic (P) approach using the Forest Growth Optimization algorithm, ranking 4th out of 5 studies [84].

The most recent benchmark, ITC 2021, introduces a new problem related to sports timetabling. This benchmark consists of 45 artificial datasets, each containing nine types of constraints, classified as either hard or soft, depending on the instance. Although it has fewer constraints compared to ITC 2011 and ITC 2019, the ITC 2021 problem is particularly challenging. Among the 13 research studies reviewed, only one managed to solve all instances under feasible conditions [86]. When focusing on studies that used heuristic methods, only one study by [87] was identified. This research employed a metaheuristic (S) approach based on Simulated Annealing and achieved an impressive rank of 2nd out of 13 studies.

V. ALGORITHM GENERALITY

Developing a universally applicable algorithm for timetabling problems is a key objective. As discussed in Section II, there are three levels of generality. The first level pertains to algorithms designed for specific datasets within a single benchmark. Most studies have successfully achieved this level of generality, particularly those reviewed in this paper. These studies not only generate feasible solutions but also optimize soft constraints across all datasets within the same benchmark. The optimization outcomes vary, as detailed in the analysis in Section IV.

TABLE 6. Comparison in ITC 2011 benchmark.

No	Researcher	Approach	Algorithm	Average Ranking
1	[78]	Hybrid (S+S)	Simulated Annealing + Iterated Local Search	1,6
2	[77]	Metaheuristic (S)	Variable Neighbourhood Search	1,91
3	[76]	Hybrid (S+S)	Simulated Annealing + Late Acceptance Hill-Climbing	2,75
4	[80]	Hyper Heuristic(S-P)	Threshold Acceptance	4,16
5	[79]	Hyper Heuristic(S-P)	Simulated Annealing	4,33
6	[81]	Hyper Heuristic(S-P)	Great Deluge	4,5

In this section, we focus on exploring the second and third levels of generality for addressing timetabling problems. We selected studies that tested their algorithms across multiple benchmarks. Table 7 provides detailed descriptions of the 20 studies we identified, including the benchmarks used, the rankings obtained from Section IV, and whether the same parameters were applied across two or more benchmarks.

For the second level of generality, which involves applying the algorithm across different benchmarks addressing the same problem, we observed two distinct cases. The first case focuses on exam timetabling using the Carter benchmark and ITC 2007 Track 1. In this context, we identified ten relevant studies. The study by [11] demonstrated strong performance on both benchmarks, ranking 3rd out of 26 studies for the Carter benchmark and 2nd out of 17 studies for the ITC 2007 Track 1 benchmark. However, this study used different parameter settings for the two benchmarks. In contrast, three other studies by [27], [28], and [30], while not achieving top results, displayed more consistent outcomes with average rankings using the same parameter settings. On the other hand, three studies showed inconsistent outcomes, performing well on one benchmark but poorly on the other. For example, the study by [21] ranked 2nd for the Carter benchmark but only 13th for ITC 2007 Track 1. Similar patterns were observed in the studies by [22], [29], and [34]. Lastly, the studies by [39] and [42] performed poorly on both benchmarks.

In the second case, we identified five studies focusing on the post-enrollment timetabling problem using both the Socha benchmark and ITC 2007 Track 2. The study by [46] performed well on both benchmarks. While it used different parameter settings for each benchmark, the parameter values were automatically determined through a preliminary procedure before optimization, supporting the concept of generality. Two other studies by [44] and [47], authored by the same researcher and employing a similar method, also demonstrated consistent and favorable results across both benchmarks. The research by [45] also delivered good results on both benchmarks, though it used different parameter settings. However, the study by [53] did not perform well on either benchmark.

For the third level of generality, where algorithms are applied to different types of timetabling problems, we observed three cases. The first case focuses on both exam and post-enrollment course timetabling, using the Carter and Socha benchmarks. We identified two studies for this case, both of which used the same parameter settings across the two benchmarks. The study by [19] performed well, ranking 6th out of 26 studies for the Carter benchmark and 8th out of 14 studies for the Socha benchmark. On the other hand, the study by [32] had slightly lower results, ranking 11th for Carter and 12th for Socha.

The second case of generality applies to exam and curriculum course timetabling, using ITC 2007 Tracks 1 and 3. We identified one study by [56], which performed well, ranking 8th out of 17 studies for ITC 2007 Track 1 and 2nd out of 7 studies for ITC 2007 Track 3. The third case focuses on post-enrollment and curriculum course timetabling. Two benchmark combinations were found. The first, using the Socha benchmark and ITC 2007 Track 3, was represented by a study from [55], which did not perform well, ranking last for the Socha benchmark and 6th out of 7 studies for ITC 2007 Track 3. The second combination, using ITC 2007 Tracks 2 and 3, was explored in a study by [68], which showed inconsistent results. The study ranked 8th out of 12 studies for ITC 2007 Track 2 but secured the top rank for ITC 2007 Track 3, finishing 1st out of 7 studies.

VI. DISCUSSION AND FURTHER RESEARCH

A. APPROACH AND ALGORITHM

In Section IV, we compared 64 studies across six benchmarks and discussed the results of three studies on the most recent benchmarks. The findings clearly indicate that single solution-based approaches are more suitable for timetabling problems, consistently achieving top ranks across all benchmarks covered in this paper, compared to population-based approaches. Among these, the metaheuristic (S) approach stands out as the most effective. It secured the top position in four benchmarks: Carter, Socha, ITC 2007 Track 1, and ITC 2007 Track 2. In the remaining two benchmarks, it still performed above average. Notably, on the newer, more challenging benchmarks, ITC 2019 and ITC 2021, this approach effectively addressed the problems, delivering particularly impressive results for ITC 2021.

In contrast, metaheuristic (P) approaches did not perform as well, appearing in only three benchmarks: Carter, Socha, and ITC 2019. This underperformance may be attributed to the nature of population-based algorithms, which typically rely on crossover operators that combine two or more solutions. In timetabling problems, maintaining feasibility while using this strategy is difficult due to the high number

TABLE 7. Comparison of the generalization level of various studies.

Researcher	Benchmarks	Rankings	Total Studies	Same Parameter Setting?
[11]	Carter ITC 2007 Track 1	3 2	26 17	No
[21]	Carter ITC 2007 Track 1	2 13	26 17	No
[22]	Carter ITC 2007 Track 1	4 16	26 17	No
[29]	Carter ITC 2007 Track 1	5 11	26 17	No
[27]	Carter ITC 2007 Track 1	7 9	26 17	Yes
[30]	Carter ITC 2007 Track 1	9 12	26 17	Yes
[42]	Carter ITC 2007 Track 1	26 14	26 17	No
[28]	Carter ITC 2007 Track 1	12 10	26 17	Yes
[39]	Carter ITC 2007 Track 1	22 17	26 17	Yes
[34]	Carter ITC 2007 Track 1	14 4	26 17	Yes
[19]	Carter Socha	6 8	26 14	Yes
[32]	Carter Socha	11 12	25 14	Yes
[46]	Socha ITC 2007 Track 2	1 2	14 11	No
[47]	Socha ITC 2007 Track 2	2 3	14 11	Yes
[44]	Socha ITC 2007 Track 2	3 4	14 11	Yes
[45]	Socha ITC 2007 Track 2	4 1	14 11	No
[53]	Socha ITC 2007 Track 2	11 7	14 11	Yes
[55]	Socha ITC 2007 Track 3	14 6	14 7	Yes
[56]	ITC 2007 Track 1 ITC 2007 Track 3	8 2	17 7	Yes
[69]	ITC 2007 Track 2 ITC 2007 Track 3	8 1	12 7	Yes

of constraints. As a result, population-based approaches are primarily applied to problems with fewer constraints, such as those of Carter and Socha. One study by [85] applied this approach to the more complex problem of ITC 2019, but provided only a brief explanation, resulting in suboptimal performance and limiting the scope for a comprehensive analysis.

Regarding hybrid approaches, several studies, including [21], [49], [56], and [78], delivered strong results. However, many other studies achieved only average outcomes. It is important to note that hybrid approaches combine different metaheuristic algorithms. Given the strong performance of single solution-based approaches, it is possible that the success of hybrid (P-S) approaches is largely driven by the single solution-based algorithm's search process. This hypothesis requires further investigation. Additionally, the hybrid (P-S) approach has not been extensively tested on benchmarks with a higher number of constraints, such as ITC 2011, possibly due to the limitations of population-based approaches in such problems.

Hyper-heuristics have shown promise in several studies, especially those employing the perturbation approach, such as [22], [59], and [69]. However, many studies using the same approach did not achieve comparable success. The construction approach, in particular, consistently underperformed and was observed only in the Carter and ITC 2007 Track 1 benchmarks. This approach focuses on constructing solutions iteratively using various combinations of low-level heuristics, which makes it challenging to produce feasible and optimally optimized solutions in problems with numerous constraints.

In terms of algorithms, Simulated Annealing has consistently delivered strong results in timetabling problems. It achieved top ranks in four benchmarks and satisfactory outcomes in the other two. As a single solution-based approach, Simulated Annealing aligns with the broader finding that single solution-based approaches are well-suited for timetabling problems. Notably, Simulated Annealing achieved the best result as the sole heuristic method in the ITC 2019 benchmark and ranked second in the ITC

2021 benchmark, where it was the only heuristic method employed.

Other single solution-based algorithms have also shown promise. The Great Deluge algorithm performed well in benchmarks such as Carter and ITC 2007 Track 1. Iterated Local Search excelled in ITC 2007 Track 3 and ITC 2011. Random Partial Neighbourhood Search yielded strong results in the Socha and ITC 2007 Track 2 benchmarks. A relatively newer algorithm, Step Counting Hill Climbing, also demonstrated great potential, securing first place in ITC 2007 Track 1.

B. GENERIC ALGORITHM

In studies on generic algorithms, certain approaches have shown promising results at levels 2 and 3 of generality. Notably, the work of [46] stands out, achieving top rankings on two different post-enrollment timetabling benchmarks using automatically set parameters. While the studies by [27] and [30] did not achieve the same level of success as [46], they consistently reported average results on two benchmarks for the exam timetabling problem. At level 3 of generality, noteworthy findings can be seen in the studies by [19] and [56], both of which demonstrated consistent performance across two distinct types of problems.

However, none of these five studies employed hyper-heuristics, which is surprising given that hyper-heuristics were developed to enhance the generality of algorithms. Studies that did incorporate hyper-heuristics, such as [22], [34], and [69], produced inconsistent results. Even more concerning were the outcomes of [39] and [42], which underperformed across all tested benchmarks. This raises an important question: whether the hyper-heuristic approach itself is unsuited for developing generic algorithms for timetabling problems or if these studies were unable to effectively apply hyper-heuristics. Further investigation is needed to clarify this issue.

C. FURTHER RESEARCH

This study provides significant insights into the timetabling problem, particularly regarding approaches and algorithms. First, single solution-based algorithms have proven to be more suitable for solving timetabling problems compared to population-based approaches. Future research should focus on developing algorithms that utilize single solution-based approaches. The metaheuristic (S) approach, identified as the most successful in this survey, should be a primary focus for further investigation. Additionally, approaches like hybrid (S + S) and hyper-heuristics with perturbation strategies have shown promise and warrant further exploration, given the limited number of studies examining these approaches. Regarding specific algorithms, Simulated Annealing has demonstrated superior performance across many studies, but other single solution-based algorithms also show potential. Future research could focus on enhancing these algorithms, either by refining existing ones or developing new approaches.

Secondly, in the area of generic algorithms, while many studies have shown commendable results, there remains room for improvement. Most studies achieve average rankings consistently, and future research could aim to improve optimization results while maintaining consistency. The preliminary run method proposed by [46] for Simulated Annealing delivered impressive results, though it was limited to post-enrollment timetabling scenarios. Future studies could apply this methodology to other timetabling problems or adapt the preliminary run approach to different algorithms. Additionally, further investigation into the role of hyper-heuristics in developing generic algorithms for timetabling is necessary.

Finally, many studies have relied on older benchmarks, some of which have been in use for 10-20 years. Future research should consider employing newer benchmarks, such as ITC 2019 and ITC 2021, to test their algorithms and ensure relevance to modern timetabling challenges.

VII. CONCLUSION

This survey offers a detailed analysis of 64 studies addressing the timetabling problem, providing key insights into the performance of various heuristic approaches across multiple benchmarks. The results indicate that single-solution-based metaheuristics, particularly Simulated Annealing, demonstrate superior performance when compared to population-based algorithms, making them the most effective for timetabling problems. While hybrid and hyper-heuristic approaches have shown promise, especially in some specific cases, their performance remains inconsistent. The development of generic algorithms capable of solving cross-domain timetabling problems is still an area requiring further research. Although significant progress has been made, challenges related to algorithmic generalizability and adaptability across different timetabling domains persist. Future research should focus on improving single-solution approaches and developing more robust comparison frameworks. Additionally, the exploration of newer benchmarks, such as ITC 2019 and ITC 2021, will be essential for advancing the field and testing the adaptability of algorithms under more complex constraints.

REFERENCES

[1] M. C. Chen, S. N. Sze, S. L. Goh, N. R. Sabar, and G. Kendall, "A survey of university course timetabling problem: Perspectives, trends and opportunities," *IEEE Access*, vol. 9, pp. 106515–106529, 2021. [Online]. Available:. https://ieeexplore.ieee.org/document/9499056/

[2] T. Song, M. Chen, Y. Xu, D. Wang, X. Song, and X. Tang, "Competition-guided multi-neighborhood local search algorithm for the university course timetabling problem," *Appl. Soft Comput.*, vol. 110, Oct. 2021, Art. no. 107624. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1568494621005457

[3] J. S. Tan, S. L. Goh, G. Kendall, and N. R. Sabar, "A survey of the state-of-the-art of optimisation methodologies in school timetabling problems," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113943. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417420307314

[4] N. Kokash, "An introduction to heuristic algorithms," Dept. Informat. Telecommun., Univ. Trento, Italy, Rep., 2005, pp. 1–8.

[5] I. G. A. Premananda, A. Tjahyanto, and A. Muklason, "Hybrid whale optimization algorithm for solving timetabling problems of ITC 2019," in *Proc. IEEE Int. Conf. Cybern. Comput. Intell. (CyberneticsCom)*, Jun. 2022, pp. 317–322.

[6] H. Babaei, J. Karimpour, and A. Hadidi, "A survey of approaches for university course timetabling problem," *Comput. Ind. Eng.*, vol. 86, pp. 43–59, Aug. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360835214003714

[7] R. A. Oude Vrielink, E. A. Jansen, E. W. Hans, and J. van Hillegersberg, "Practices in timetabling in higher education institutions: A systematic review," *Ann. Oper. Res.*, vol. 275, no. 1, pp. 145–160, Apr. 2019. [Online]. Available: http://link.springer.com/10.1007/s10479-017-2688-8

[8] A. Bashab, A. O. Ibrahim, E. E. AbedElgabar, M. A. Ismail, A. Elsafi, A. Ahmed, and A. Abraham, "A systematic mapping study on solving university timetabling problems using meta-heuristic algorithms," *Neural Comput. Appl.*, vol. 32, no. 23, pp. 17397–17432, Dec. 2020. [Online]. Available: https://link.springer.com/10.1007/s00521-020-05110-3

[9] S. Ceschia, L. Di Gaspero, and A. Schaerf, "Educational timetabling: Problems, benchmarks, and state-of-the-art results," *Eur. J. Oper. Res.*, vol. 308, no. 1, pp. 1–18, Jul. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221722005641

[10] N. Pillay, "Incorporating chaos into the developmental approach for solving the examination timetabling problem," *Int. J. Bio-Inspired Comput.*, vol. 8, no. 6, p. 355, 2016. [Online]. Available: http://www.inderscience.com/link.php?id=81327

[11] E. K. Burke and Y. Bykov, "An adaptive flex-deluge approach to university exam timetabling," *INFORMS J. Comput.*, vol. 28, no. 4, pp. 781–794, Nov. 2016. [Online]. Available: http://pubsonline.informs.org/doi/10.1287/ijoc.2015.0680

[12] D. B. Gümüş, E. Özcan, J. Atkin, and J. H. Drake, "An investigation of F-race training strategies for cross domain optimisation with memetic algorithms," *Inf. Sci.*, vol. 619, pp. 153–171, Jan. 2023.

[13] N. Pillay, "A survey of school timetabling research," *Ann. Oper. Res.*, vol. 218, no. 1, pp. 261–293, Jul. 2014.

[14] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Comput. Ind. Eng.*, vol. 137, Nov. 2019, Art. no. 106040.

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

[16] E. K. Burke, M. R. Hyde, G. Kendall, G. Ochoa, E. Özcan, and J. R. Woodward, "A classification of hyper-heuristic approaches: Revisited," in *Handbook of Metaheuristics* (International Series in Operations Research and Management Science). Cham, Switzerland: Springer, 2019, pp. 453–477.

[17] N. Pillay, "A review of hyper-heuristics for educational timetabling," *Ann. Oper. Res.*, vol. 239, no. 1, pp. 3–38, Apr. 2016.

[18] M. W. Carter, G. Laporte, and S. Y. Lee, "Examination timetabling: Algorithmic strategies and applications," *J. Oper. Res. Soc.*, vol. 47, no. 3, pp. 373–383, Mar. 1996.

[19] C. W. Fong, H. Asmuni, B. McCollum, P. Mcmullan, and S. Omatu, "A new hybrid imperialist swarm-based optimization algorithm for university timetabling problems," *Inf. Sci.*, vol. 283, pp. 1–21, Nov. 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025514005994

[20] R. Bellio, S. Ceschia, L. Di Gaspero, and A. Schaerf, "Two-stage multi-neighborhood simulated annealing for uncapacitated examination timetabling," *Comput. Oper. Res.*, vol. 132, Aug. 2021, Art. no. 105300. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0305054821000927

[21] N. Leite, C. M. Fernandes, F. Melício, and A. C. Rosa, "A cellular memetic algorithm for the examination timetabling problem," *Comput. Oper. Res.*, vol. 94, pp. 118–138, Jun. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0305054818300455

[22] P. Demeester, B. Bilgin, P. De Causmaecker, and G. V. Berghe, "A hyper-heuristic approach to examination timetabling problems: Benchmarks and a new problem from practice," *J. Scheduling*, vol. 15, no. 1, pp. 83–103, Feb. 2012. [Online]. Available: http://link.springer.com/10.1007/s10951-011-0258-5

[23] M. A. Al-Betar, "A β-hill climbing optimizer for examination timetabling problem," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 653–666, Jan. 2021. [Online]. Available: https://link.springer.com/10.1007/s12652-020-02047-2

[24] T. C. Pais and P. Amaral, "Managing the Tabu list length using a fuzzy inference system: An application to examination timetabling," *Ann. Oper. Res.*, vol. 194, no. 1, pp. 341–363, Apr. 2012. [Online]. Available: http://link.springer.com/10.1007/s10479-011-0867-6

[25] J. Li, R. Bai, Y. Shen, and R. Qu, "Search with evolutionary ruin and stochastic rebuild: A theoretic framework and a case study on exam timetabling," *Eur. J. Oper. Res.*, vol. 242, no. 3, pp. 798–806, May 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0377221714009060

[26] N. Pillay and E. Özcan, "Automated generation of constructive ordering heuristics for educational timetabling," *Ann. Oper. Res.*, vol. 275, no. 1, pp. 181–208, Apr. 2019. [Online]. Available: http://link.springer.com/10.1007/s10479-017-2625-x

[27] M. Alzaqebah and S. Abdullah, "Hybrid bee colony optimization for examination timetabling problems," *Comput. Oper. Res.*, vol. 54, pp. 142–154, Feb. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0305054814002524

[28] S. Abdullah and M. Alzaqebah, "A hybrid self-adaptive bees algorithm for examination timetabling problems," *Appl. Soft Comput.*, vol. 13, no. 8, pp. 3608–3620, Aug. 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1568494613001294

[29] A. K. Mandal, M. N. M. Kahar, and G. Kendall, "Addressing examination timetabling problem using a partial exams approach in constructive and improvement," *Computation*, vol. 8, no. 2, p. 46, May 2020. [Online]. Available: https://www.mdpi.com/2079-3197/8/2/46

[30] M. Alzaqebah and S. Abdullah, "An adaptive artificial bee colony and late-acceptance hill-climbing algorithm for examination timetabling," *J. Scheduling*, vol. 17, no. 3, pp. 249–262, Jun. 2014. [Online]. Available: http://link.springer.com/10.1007/s10951-013-0352-y

[31] M. A. Al-Betar, A. T. Khader, and I. A. Doush, "Memetic techniques for examination timetabling," *Ann. Oper. Res.*, vol. 218, no. 1, pp. 23–50, Jul. 2014. [Online]. Available: http://link.springer.com/10.1007/s10479-013-1500-7

[32] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "A honey-bee mating optimization algorithm for educational timetabling problems," *Eur. J. Oper. Res.*, vol. 216, no. 3, pp. 533–543, Feb. 2012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0377221711007181

[33] B. A. Aldeeb, M. A. Al-Betar, N. M. Norwawi, K. A. Alissa, M. K. Alsmadi, A. A. Hazaymeh, and M. Alzaqebah, "Hybrid intelligent water drops algorithm for examination timetabling problem," *J. King Saud Univ., Comput. Inf. Sci.*, vol. 34, no. 8, pp. 4847–4859, Sep. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1319157821001634

[34] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "Grammatical evolution hyper-heuristic for combinatorial optimization problems," *IEEE Trans. Evol. Comput.*, vol. 17, no. 6, pp. 840–861, Dec. 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6595625/

[35] R. Qu, N. Pham, R. Bai, and G. Kendall, "Hybridising heuristics within an estimation distribution algorithm for examination timetabling," *Appl. Intell.*, vol. 42, no. 4, pp. 679–693, Jun. 2015. [Online]. Available: http://link.springer.com/10.1007/s10489-014-0615-0

[36] N. Pillay, "Evolving hyper-heuristics for the uncapacitated examination timetabling problem," *J. Oper. Res. Soc.*, vol. 63, no. 1, pp. 47–58, Jan. 2012. [Online]. Available: https://www.tandfonline.com/doi/full/10.1057/jors.2011.12

[37] Y. Lei, M. Gong, L. Jiao, J. Shi, and Y. Zhou, "An adaptive coevolutionary memetic algorithm for examination timetabling problems," *Int. J. Bio-Inspired Comput.*, vol. 10, no. 4, p. 248, 2017. [Online]. Available: http://www.inderscience.com/link.php?id=87918

[38] M. A. Ahandani, M. T. V. Baghmisheh, M. A. B. Zadeh, and S. Ghaemi, "Hybrid particle swarm optimization transplanted into a hyper-heuristic structure for solving examination timetabling problem," *Swarm Evol. Comput.*, vol. 7, pp. 21–34, Dec. 2012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2210650212000417

[39] E. K. Burke, R. Qu, and A. Soghier, "Adaptive selection of heuristics for improving exam timetables," *Ann. Oper. Res.*, vol. 218, no. 1, pp. 129–145, Jul. 2014. [Online]. Available: http://link.springer.com/10.1007/s10479-012-1140-3

[40] X. Hao, R. Qu, and J. Liu, "A unified framework of graph-based evolutionary multitasking hyper-heuristic," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 35–47, Feb. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9084121/

[41] Y. Lei, M. Gong, L. Jiao, and Y. Zuo, "A memetic algorithm based on hyper-heuristics for examination timetabling problems," *Int. J. Intell. Comput. Cybern.*, vol. 8, no. 2, pp. 139–151, Jun. 2015. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/IJICC-02-2015-0005/full/html

[42] N. R. Sabar, M. Ayob, R. Qu, and G. Kendall, "A graph coloring constructive hyper-heuristic for examination timetabling problems," *Appl. Intell.*, vol. 37, no. 1, pp. 1–11, Jul. 2012. [Online]. Available: http://link.springer.com/10.1007/s10489-011-0309-9

[43] K. Socha, M. Sampels, and M. Manfrin, "Ant algorithms for the university course timetabling problem with regard to the state-of-the-art," in *Proc. Workshops on Applications of Evolutionary Computation*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 2611, 2003, pp. 334–345.

[44] S. L. Goh, G. Kendall, and N. R. Sabar, "Improved local search approaches to solve the post enrolment course timetabling problem," *Eur. J. Oper. Res.*, vol. 261, no. 1, pp. 17–29, Aug. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0377221717300759

[45] Y. Nagata, "Random partial neighborhood search for the post-enrollment course timetabling problem," *Comput. Oper. Res.*, vol. 90, pp. 84–96, Feb. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0305054817302447

[46] S. L. Goh, G. Kendall, N. R. Sabar, and S. Abdullah, "An effective hybrid local search approach for the post enrolment course timetabling problem," *OPSEARCH*, vol. 57, no. 4, pp. 1131–1163, Dec. 2020. [Online]. Available: https://link.springer.com/10.1007/s12597-020-00444-x

[47] S. L. Goh, G. Kendall, and N. R. Sabar, "Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem," *J. Oper. Res. Soc.*, vol. 70, no. 6, pp. 873–888, Jun. 2019. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/01605682.2018.1468862

[48] R. Bai, J. Blazewicz, E. K. Burke, G. Kendall, and B. McCollum, "A simulated annealing hyper-heuristic methodology for flexible decision support," *4OR*, vol. 10, no. 1, pp. 43–66, Mar. 2012. [Online]. Available: http://link.springer.com/10.1007/s10288-011-0182-8

[49] R. A. Aziz, M. Ayob, Z. Othman, Z. Ahmad, and N. R. Sabar, "An adaptive guided variable neighborhood search based on honey-bee mating optimization algorithm for the course timetabling problem," *Soft Comput.*, vol. 21, no. 22, pp. 6755–6765, Nov. 2017. [Online]. Available: http://link.springer.com/10.1007/s00500-016-2225-8

[50] A. Abuhamdah, M. Ayob, G. Kendall, and N. R. Sabar, "Population based local search for university course timetabling problems," *Appl. Intell.*, vol. 40, no. 1, pp. 44–53, Jan. 2014. [Online]. Available: http://link.springer.com/10.1007/s10489-013-0444-6

[51] M. A. Al-Betar, A. T. Khader, and M. Zaman, "University course timetabling using a hybrid harmony search metaheuristic algorithm," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 5, pp. 664–681, Sep. 2012. [Online]. Available: http://ieeexplore.ieee.org/document/6135818/

[52] A. L. Bolaji, A. T. Khader, M. A. Al-Betar, and M. A. Awadallah, "University course timetabling using hybridized artificial bee colony with Hill climbing optimizer," *J. Comput. Sci.*, vol. 5, no. 5, pp. 809–818, Sep. 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877750314000441

[53] G. Jaradat, M. Ayob, and Z. Ahmad, "On the performance of scatter search for post-enrolment course timetabling problems," *J. Combinat. Optim.*, vol. 27, no. 3, pp. 417–439, Apr. 2014. [Online]. Available: http://link.springer.com/10.1007/s10878-012-9521-8

[54] A. Rezaeipanah, S. S. Matoori, and G. Ahmadi, "A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search," *Appl. Intell.*, vol. 51, no. 1, pp. 467–492, Jan. 2021. [Online]. Available: https://link.springer.com/10.1007/s10489-020-01833-x

[55] S. Abdullah, H. Turabieh, B. McCollum, and P. Mcmullan, "A hybrid metaheuristic approach to the university course timetabling problem," *J. Heuristics*, vol. 18, no. 1, pp. 1–23, Feb. 2012. [Online]. Available: https://link.springer.com/10.1007/s10732-010-9154-y

[56] S. Abdullah and H. Turabieh, "On the use of multi neighbourhood structures within a Tabu-based memetic approach to university timetabling problems," *Inf. Sci.*, vol. 191, pp. 146–168, May 2012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025511006670

[57] Y. Bykov and S. Petrovic, "A step counting Hill climbing algorithm applied to university examination timetabling," *J. Scheduling*, vol. 19, no. 4, pp. 479–492, Aug. 2016. [Online]. Available: http://link.springer.com/10.1007/s10951-016-0469-x

[58] C. Gogos, P. Alefragis, and E. Housos, "An improved multi-staged algorithmic process for the solution of the examination timetabling problem," *Ann. Oper. Res.*, vol. 194, no. 1, pp. 203–221, Apr. 2012. [Online]. Available: http://link.springer.com/10.1007/s10479-010-0712-3

[59] N. R. Sabar and G. Kendall, "Population based Monte Carlo tree search hyper-heuristic for combinatorial optimization problems," *Inf. Sci.*, vol. 314, pp. 225–239, Sep. 2015.

[60] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "A dynamic multiarmed bandit-gene expression programming hyper-heuristic for combinatorial optimization problems," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 217–228, Feb. 2015. [Online]. Available: http://ieeexplore.ieee.org/document/6824192/

[61] M. Battistutta, A. Schaerf, and T. Urli, "Feature-based tuning of single-stage simulated annealing for examination timetabling," *Ann. Oper. Res.*, vol. 252, no. 2, pp. 239–254, May 2017. [Online]. Available: http://link.springer.com/10.1007/s10479-015-2061-8

[62] N. Leite, F. Melício, and A. C. Rosa, "A fast simulated annealing algorithm for the examination timetabling problem," *Expert Syst. Appl.*, vol. 122, pp. 137–151, May 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417418308169

[63] A. Soghier and R. Qu, "Adaptive selection of heuristics for assigning time slots and rooms in exam timetables," *Appl. Intell.*, vol. 39, no. 2, pp. 438–450, Sep. 2013. [Online]. Available: http://link.springer.com/10.1007/s10489-013-0422-z

[64] R. Lewis, "A survey of metaheuristic-based techniques for university timetabling problems," *OR Spectr.*, vol. 30, no. 1, pp. 167–190, Nov. 2007. [Online]. Available: http://link.springer.com/10.1007/s00291-007-0097-0

[65] C. Nothegger, A. Mayer, A. Chwatal, and G. R. Raidl, "Solving the post enrolment course timetabling problem by ant colony optimization," *Ann. Oper. Res.*, vol. 194, no. 1, pp. 325–339, Apr. 2012. [Online]. Available: http://link.springer.com/10.1007/s10479-012-1078-5

[66] R. Lewis, "A time-dependent metaheuristic algorithm for post enrolment-based course timetabling," *Ann. Oper. Res.*, vol. 194, no. 1, pp. 273–289, Apr. 2012. [Online]. Available: http://link.springer.com/10.1007/s10479-010-0696-z

[67] S. Ceschia, L. Di Gaspero, and A. Schaerf, "Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem," *Comput. Oper. Res.*, vol. 39, no. 7, pp. 1615–1624, Jul. 2012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0305054811002759

[68] J. A. Soria-Alcaraz, G. Ochoa, M. A. Sotelo-Figeroa, and E. K. Burke, "A methodology for determining an effective subset of heuristics in selection hyper-heuristics," *Eur. J. Oper. Res.*, vol. 260, no. 3, pp. 972–983, Aug. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0377221717300772

[69] J. A. Soria-Alcaraz, E. Özcan, J. Swan, G. Kendall, and M. Carpio, "Iterated local search using an add and delete hyper-heuristic for university course timetabling," *Appl. Soft Comput.*, vol. 40, pp. 581–593, Mar. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1568494615007760

[70] J. A. Soria-Alcaraz, G. Ochoa, J. Swan, M. Carpio, H. Puga, and E. K. Burke, "Effective learning hyper-heuristics for the course timetabling problem," *Eur. J. Oper. Res.*, vol. 238, no. 1, pp. 77–86, Oct. 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0377221714002859

[71] L. Di Gaspero, B. McCollum, and A. Schaerf, "The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3)," Queen's Univ., Belfast, U.K., Queen's, Rep., 2007.

[72] M. J. Geiger, "Applying the threshold accepting metaheuristic to curriculum based course timetabling," *Ann. Oper. Res.*, vol. 194, no. 1, pp. 189–202, Apr. 2012. [Online]. Available: http://link.springer.com/10.1007/s10479-010-0703-4

[73] A. Kiefer, R. F. Hartl, and A. Schnell, "Adaptive large neighborhood search for the curriculum-based course timetabling problem," *Ann. Oper. Res.*, vol. 252, no. 2, pp. 255–282, May 2017. [Online]. Available: http://link.springer.com/10.1007/s10479-016-2151-2

[74] R. Bellio, L. Di Gaspero, and A. Schaerf, "Design and statistical analysis of a hybrid local search algorithm for course timetabling," *J. Scheduling*, vol. 15, no. 1, pp. 49–61, Feb. 2012. [Online]. Available: http://link.springer.com/10.1007/s10951-011-0224-2
[75] G. Post, L. Di Gaspero, J. H. Kingston, B. McCollum, and A. Schaerf, "The third international timetabling competition," *Ann. Oper. Res.*, vol. 239, no. 1, pp. 69–75, Apr. 2016.
[76] G. H. G. Fonseca, H. G. Santos, and E. G. Carrano, "Late acceptance hill-climbing for high school timetabling," *J. Scheduling*, vol. 19, no. 4, pp. 453–465, Aug. 2016. [Online]. Available: http://link.springer.com/10.1007/s10951-015-0458-5
[77] G. H. G. Fonseca and H. G. Santos, "Variable neighborhood search based algorithms for high school timetabling," *Comput. Oper. Res.*, vol. 52, pp. 203–208, Dec. 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0305054813003328
[78] G. H. G. da Fonseca, H. G. Santos, T. Â. M. Toffolo, S. S. Brito, and M. J. F. Souza, "GOAL solver: A hybrid local search based solver for high school timetabling," *Ann. Oper. Res.*, vol. 239, no. 1, pp. 77–97, Apr. 2016. [Online]. Available: http://link.springer.com/10.1007/s10479-014-1685-4
[79] M. Kalender, A. Kheiri, E. Özcan, and E. K. Burke, "A greedy gradient-simulated annealing selection hyper-heuristic," *Soft Comput.*, vol. 17, no. 12, pp. 2279–2292, Dec. 2013. [Online]. Available: http://link.springer.com/10.1007/s00500-013-1096-5
[80] A. Kheiri, E. Özcan, and A. J. Parkes, "A stochastic local search algorithm with adaptive acceptance for high-school timetabling," *Ann. Oper. Res.*, vol. 239, no. 1, pp. 135–151, Apr. 2016. [Online]. Available: http://link.springer.com/10.1007/s10479-014-1660-0
[81] L. N. Ahmed, E. Özcan, and A. Kheiri, "Solving high school timetabling problems worldwide using selection hyper-heuristics," *Expert Syst. Appl.*, vol. 42, no. 13, pp. 5463–5471, Aug. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417415001670
[82] T. Müller, H. Rudová, and Z. Müllerová, "University course timetabling and international timetabling competition 2019," in *Proc. 12th Int. Conf. Pract. Theory Automated Timetabling (PATAT)*, vol. 1, 2018, pp. 5–31.
[83] K. Sylejmani, E. Gashi, and A. Ymeri, "Simulated annealing with penalization for university course timetabling," *J. Scheduling*, vol. 26, no. 5, pp. 497–517, Oct. 2023.
[84] S. Abdipoor, R. Yaakob, S. L. Goh, S. Abdullah, K. A. Kasmiran, and H. Hamdan, "International timetabling competition 2019: A systematic literature review," in *Proc. IEEE Int. Conf. Autom. Control Intell. Syst. (I2CACIS)*, Jun. 2023, pp. 22–27.
[85] K. Er-rhaimini, "Forest growth optimization for solving timetabling problems," in *Proc. Int. Timetabling Competition*, 2019, pp. 1–4.
[86] D. Van Bulck and D. Goossens, "The international timetabling competition on sports timetabling (ITC2021)," *Eur. J. Oper. Res.*, vol. 308, no. 3, pp. 1249–1267, Aug. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221722009201
[87] R. M. Rosati, M. Petris, L. Di Gaspero, and A. Schaerf, "Multi-neighborhood simulated annealing for the sports timetabling competition ITC2021," *J. Scheduling*, vol. 25, no. 3, pp. 301–319, Jun. 2022.

I. GUSTI AGUNG PREMANANDA received the Bachelor of Computer Science and master's degrees in information systems from Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree in information systems.

ARIS TJAHYANTO received the bachelor's degree from the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember (ITS), in 1989, the Master's in Computer Studies (M.Com.) degree from the Faculty of Computer Science, University of Indonesia (UI), in 1995, and the Ph.D. degree in electrical engineering (artificial intelligence for sound recognition) from ITS, in 2015. Starting in 1991, he was a Lecturer with ITS, where he is currently with the Data Acquisition and Information Dissemination Laboratory (ADDI), ITS Information Systems Department, with a focus on research related to artificial intelligence. He is teaching courses, including business analytics and system development and implementation.

AHMAD MUKLASON received the Bachelor of Computer Science degree from Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, in 2006, the Master of Science degree from the Computer and Information Sciences Department, Universiti Teknologi Petronas, Malaysia, in 2009, and the Ph.D. degree from the School of Computer Sciences, The University of Nottingham, U.K., in 2017. He is currently an Associate Professor with the Data Engineering and Business Intelligence Laboratory, Department of Information Systems, Faculty of Intelligent Electrical, Electronic Engineering and Informatics, Institut Teknologi Sepuluh Nopember.