IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 April 2025, accepted 20 April 2025, date of publication 29 April 2025, date of current version 6 May 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3565522

RESEARCH ARTICLE

Optimality Versus Generality: Performance Assessment of Meta-Heuristics in Educational Timetabling

SINA ABDIPOOR[1], RAZALI YAAKOB[1], SAY LENG GOH[2], SALWANI ABDULLAH[3], HAZLINA HAMDAN[1], AND KHAIRUL AZHAR KASMIRAN[1]

[1]Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM), Serdang, Selangor 43400, Malaysia
[2]Optimization and Visual Analytics Research Group, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Labuan International Campus, Labuan 87000, Malaysia
[3]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi, Selangor 43600, Malaysia

Corresponding author: Razali Yaakob (razaliy@upm.edu.my)

ABSTRACT Educational timetabling, a principal branch of operations research, presents challenging combinatorial optimization problems widely encountered in educational institutions. Meta-heuristics have commonly been applied to these problems and managed to attain promising performance in terms of optimality. However, their general applicability has been overlooked, hindering their effectiveness as versatile solvers. The limited generalizability of current approaches is the primary hurdle between the literature and real-world applications. This paper addresses this gap by introducing a generality taxonomy and conducting comprehensive theoretical and empirical analyses. This study highlights the adverse impact of extreme parameter tuning on generality, emphasizing the need for more generalized approaches. Furthermore, it introduces a performance assessment framework, penalizing problem-tailored solutions. It also examines the optimality vs. generality performance of the state-of-the-art approaches of the latest university course timetabling benchmark to further reinforce our claim and validate the efficacy of our framework. Our findings indicate that the current literature prioritizes optimality over generality. We believe adopting the proposed assessment framework is crucial for bridging the gap between research and practical applications, enabling fairer comparisons, and encouraging more adaptable approaches.

INDEX TERMS Generality, meta-heuristic, operations research, optimality, university course timetabling.

I. INTRODUCTION

The Educational Timetabling Problem (ETP) is a major area of research in the Operations Research (OR) field [1], [2] that has drawn much attention in recent years [3], [4], [5]. This Combinatorial Optimization Problem (COP) aims to schedule courses subject to a set of predefined constraints in order to maximize utilization and satisfy stakeholders' requirements [6], [7]. ETP can be classified into university and (high) school timetabling problems [8]. Figure 1 illustrates the general problem scheme of ETP.

As the main branch of ETP, the University Course Timetabling Problem (UCTP) involves allocating a physical room, a time period, and a list of students to classes while adhering to all hard constraints and minimizing soft constraint violations [9], [10]. The objective of this Non-deterministic Polynomial-time hard (NP-hard) problem is to construct a feasible schedule while maximizing efficiency [11]. The approaches to this problem in the literature can be categorized into OR-based techniques, meta-heuristics, hyper-heuristics, multi-criteria/objective, and hybrid approaches [1].

Meta-heuristics are iterative processes directing subordinate heuristics [12]. They are capable of exploring large solution spaces and tackling a variety of different problems as

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

FIGURE 1. The general problem scheme for educational timetabling problems.

they make relatively few assumptions about the problem [13], [14]. These approaches have often been utilized in the literature, achieving superior results on benchmark datasets, and are the most common methods in practice [5], [15].

Despite significant advances in the literature, course timetabling is still being conducted manually in many educational institutions. Manual timetabling is imprecise, time-consuming, and can result in significant wastage of resources [4], [6], [16]. Bridging the gap between the literature and real-world applications of the UCTP remains a major challenge in this domain [8]. This gap can be attributed to the discrepancy between different definitions of the UCTP and the lack of generalizability of existing solvers [17]. In spite of considerable efforts towards unifying the definition of the UCTP, through the organization of the International Timetabling Competition (ITC) by Practice and Theory of Automated Timetabling (PATAT) and the Meta-heuristic Network (MN), the evaluation of solver generality has been overlooked, with methods solely assessed based on optimality [18]. As achieving new Best Known Solutions (BKS) is essential for successful publications of novel methods in top-tier journals, many researchers have resorted to problem-tailored encoding and extreme parameter tuning to achieve BKS on specific instances. This often compromises the consistency of their method's performance across other instances and datasets. This issue is particularly critical for meta-heuristics and hyper-heuristics, which aim to serve as general problem solvers. Hence, a comprehensive investigation of solver generality is imperative.

The issue of generality is crucial within the context of UCTP. We believe this issue to be the major contributing factor preventing literature methods from being adopted in real-world applications. A method with high optimality, performing exceptionally well on a specific set of instances and achieving a new BKS, may be desirable in theoretical contexts. However, if it performs poorly on other instances, it becomes unsuitable for real-world applications. Consider a solver that excels only for a specific university, a single faculty, or even certain semesters when the problem size is unusually large or small. Such inconsistency makes it impractical for real-world adoption. Currently, this performance consistency (generality) is overlooked in the performance evaluation of ETP solvers in the literature. We believe that incorporating a generality metric into performance assessment is crucial for future progress, as it can encourage the adoption of literature methods in real-world UCTP applications.

This paper aims to address this challenge by introducing a novel performance evaluation framework for measuring the efficiency of meta-heuristics on ETPs, both in terms of optimality and generality, in order to discourage problem-tailored strategies and excessive parameter tuning.

The main contributions of this paper are:

1) Investigating the lack of generality in existing methods as a key factor resulting in the current gap between the literature and real-world applications of the ETP.
2) Introducing a comprehensive taxonomy for categorizing optimality, generality, and their respective levels.
3) Conducting a thorough theoretical analysis to study the performance trade-off between optimality and generality in meta-heuristics.
4) Proposing a novel performance measurement framework for assessing meta-heuristics, considering both optimality and generality.
5) Applying our proposed evaluation framework to UCTP state-of-the-art approaches on the latest benchmark, examining their optimality and generality performance to demonstrate the unfavorable effects of extreme parameter tuning and showcase the efficacy of our framework.

The rest of this paper is organized as follows. Section II defines the problem and introduces the concepts of optimality and generality. Section III provides a summary of previous works on ETP, UCTP, and generality. The theoretical analysis exploring the trade-off between optimality and generality is presented in Section IV. Our proposed framework is presented in Section V, and its effectiveness is demonstrated through empirical experimentations in Section VI. Finally, Section VII concludes our findings and suggests possible future work opportunities.

II. RESEARCH BACKGROUND

To provide the necessary background, this section describes the terms used in this research, defines the concept of optimality, and presents a classification scheme for generality.

A. TERMINOLOGY

Problem instance represents an instance in a benchmark dataset. In the case of UCTP, the ITC 2019 dataset comprises 30 real-world samples gathered from ten institutions across five continents [19]. Each sample is characterized by a unique set of features, including the number of courses and students, and is referred to as a problem instance.

Benchmark dataset refers to a group of problem instances. These instances can originate from real-world scenarios or be generated artificially [17]. A benchmark dataset consists of a set of problem instances, a collection

of established hard and soft constraints, and a cost function. Alongside other dataset-specific attributes, such as the time limit, these features aim to introduce a general principle over all problem instances to simplify the application of a solver to all instances. In the case of UCTP, ITC 2007, ITC 2002, Socha, Hard, and ITC 2019 are the most common datasets [8]. Since the performance of a method on a single problem instance may not be representative, benchmark datasets provide a more comprehensive measure of a method's performance.

Problem refers to a topic within a specific field, often centered around a distinct task. While different problems within a field can have unique attributes or constraints, they generally follow similar problem representations governed by the problem field. As depicted in Figure 1, Course Timetabling and Examination Timetabling [2] are the two problems within the UCTP problem field.

Problem field contains a group of problems centered around a common subject. These problems often share their general objective but differ in their features due to the distinct characteristics of the fields they derive from. In the domain of ETP, university and school timetabling [4] are the main problem fields (refer to Figure 1). While both of these problem fields deal with timetabling in academic settings, the attributes of school timetabling are often different from those in universities.

Problem domain is defined as a group of problem fields on a specific matter. Domains in OR are predominantly inspired by challenges faced in real-world scenarios [18]. For example, ETP [3], Vehicle Routing Problem (VRP) [20], Job-Shop Scheduling Problem (JSSP) [21], and Nurse Rostering Problem (NRP) [22] are some popular COPs in Scheduling.

Figure 2 encapsulates these terminologies, summarising the distinctions and providing illustrative examples.

B. OPTIMALITY

Optimality (also known as generality level 0) refers to the best mathematically proven solution for a given problem instance [12]. The definition of ''best'' is problem-specific, determined by the dataset's constraints and cost function. Optimality persists in finding the best-quality solution(s) to a single (or a group of) problem instance(s) in a given dataset. For a single problem instance, it evaluates a method's effectiveness based on the cost or fitness function, without considering its performance across other instances. This measurement defines the Best Known Solution (BKS) and is commonly reported in the literature as an indicator of a proposed method's performance.

C. GENERALITY

The notion of generality was first introduced by McCarthy [23]. Unlike optimality, which strives to discover the best solution to a specific problem instance, generality deals with performance consistency and widespread adaptability of a solver across different problem instances, datasets, domains, and fields [23]. Similarly, the generalizability of a method can be quantified as the amount of effort and changes needed to adapt it to a new problem. Pillay et al. [18] presented generality and its different levels for hyper-heuristics.

- **Generality level 1** acts over all problem instances in a single benchmark dataset. For example, in the case of UCTP, a method with high level 1 generality can be effortlessly applied to all 30 problem instances of the ITC 2019 while maintaining consistent performance. This is the lowest and most commonly observed level of generality for meta-heuristics in the literature [3], [5], as handling level 0 can be accomplished using direct approaches (deterministic) or problem-specific heuristics.
- **Generality level 2** deals with different benchmark datasets of a single problem. For instance, a method with high level 2 generality is capable of addressing ITC 2019, ITC 2007, Socha, etc., and obtaining consistent performance. Meta-heuristics are able to attain this level of generality, albeit with minor changes required to adjust to different problem representations. Table 1 presents examples of meta-heuristics with level 2 generality in the UCTP.
- **Generality level 3a** refers to the ability to address distinct problems within a problem field. In the context of UCTP, a method exhibiting a level 3a generality would be able to address both university course timetabling and university examination timetabling with consistent performance. While meta-heuristics, theoretically, retain the potential to achieve this level of generality, hyper-heuristics are notably more effective in achieving level 3a generality due to their higher overall generalizability [18], [24], [25], [26], [27], [28].
- **Generality level 3b** indicates the ability of a method to maintain consistent performance across diverse problem fields within a problem domain. An approach demonstrating level 3b generality would be able to simultaneously address the university and school timetabling problem fields in the ETP domain. This level of generality is particularly difficult to achieve as different problem fields may have subtle differences in their problem definitions and objectives. Hyper-heuristics have been predominantly employed in the literature to achieve this level of generality [18], [29], [30], [31].
- **Generality level 4** works across multiple problem domains without being tailored to a specific one. For example, a method with level 4 generality would be able to handle VRP, JSSP, and NRP. In the existing literature, only a few hyper-heuristics have managed to reach this level of generality [18], [32], [33], [34], [35].

Table 2 summarizes the generality taxonomy presented in this section.

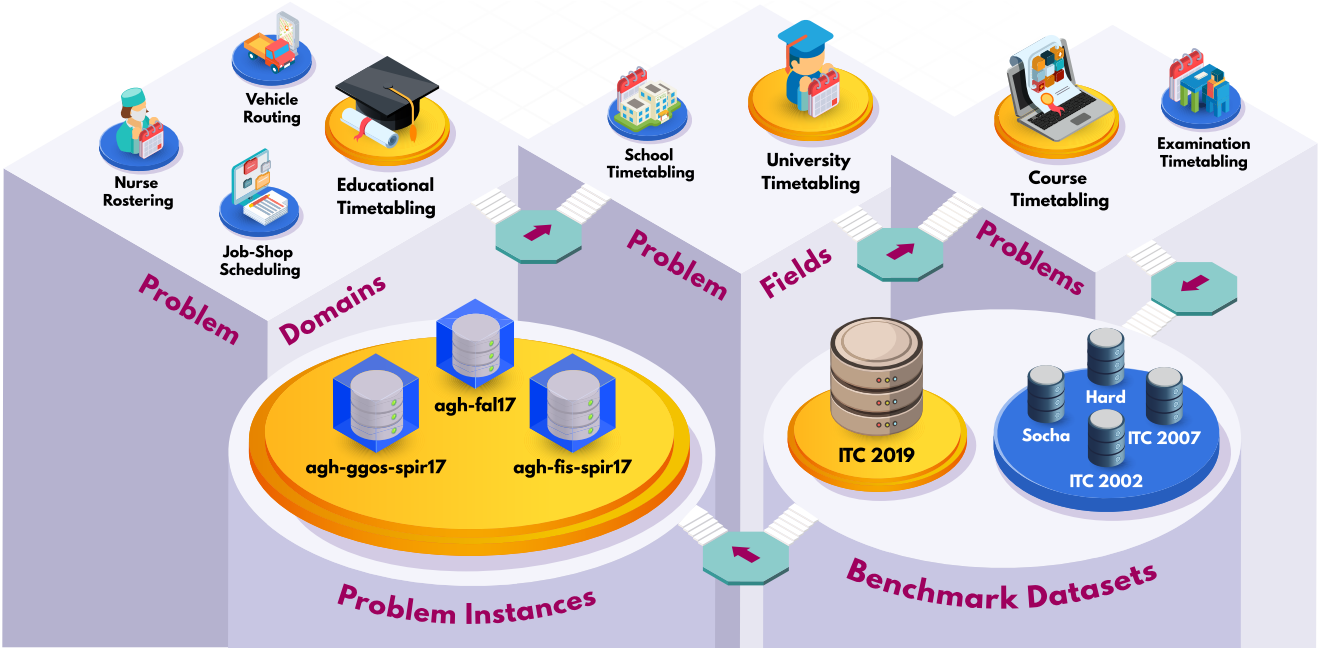


FIGURE 2. Encapsulation of terminologies.

TABLE 1. Examples of meta-heuristic approaches from the literature that exhibit Level 2 generality in solving the university course timetabling problem.

Author	Method	Category	Datasets
Badoni et al. [36]	Genetic Algorithm	Population-based	ITC 2007, Socha
Chen et al. [37]	Local Search	Single Solution-based	ITC 2007, Real-World
Rezaeipanah et al. [38]	Genetic Algorithm	Population-based	ITC 2007, BenPaechter
Goh et al. [6]	Simulated Annealing	Single Solution-based	ITC 2007, ITC 2002, Hard, Socha
Gozali et al. [39]	Genetic Algorithm	Population-based	ITC 2007, Real-World
Goh et al. [40]	Simulated Annealing	Single Solution-based	ITC 2007, ITC 2002, Socha
Goh et al. [10]	Simulated Annealing	Single Solution-based	ITC 2007, ITC 2002, Socha
Fong et al. [41]	Artificial Bee Colony	Population-based	Socha, Carter's

TABLE 2. A summarized taxonomy of different generality levels.

Generality	Domain	Common Solvers
Level 0 (Optimality)	Single Problem Instance, Single Benchmark Dataset	Direct Approaches, Heuristics
Level 1	Different Problem Instances, Single Benchmark Dataset	Meta-heuristics
Level 2	Different Benchmark Datasets, Same Problem	Meta-heuristics, Hyper-heuristics
Level 3a	Different Problems, Same Problem Field	Meta-heuristics, Hyper-heuristics
Level 3b	Different Problem Fields, Same Problem Domain	Hyper-heuristics
Level 4	Different Problem Domains	Hyper-heuristics

III. RELATED WORK

In operations research, optimization plays a significant role, often focusing on maximizing resource utilization to minimize costs and improve efficiency [42]. Resource scheduling, a key subproblem within optimization, frequently involves timetabling, which has attracted considerable attention from the OR research community [2].

As one of the most dynamic OR domains, ETP has been extensively studied in the literature. Ceschia et al. [3] surveyed this problem and identified six formulations of its subproblems. They presented standard representations and common benchmark datasets for each of these subproblems. Through a comprehensive empirical comparison, they revealed that most State-Of-The-Art (SOTA) approaches in this domain are based on meta-heuristics. It was shown that the UCTP has noticeably drawn more attention compared to school timetabling. This study also shed light on the prevailing challenges faced in the ETP domain. The lack of generality, due to different problem definitions and a lack of basis for fair comparison of solvers' performance, was reported as a significant observation in this work. They argued that there is a need for a shift towards generality, which can be achieved by establishing standard definitions for the ETP subproblems, providing explicit competition grounds for a fair comparison, avoiding excessive fine-tuning, and facilitating experiment replication by making solutions/source codes available.

Chen et al. [1] provided a comprehensive overview of the UCTP. They reviewed the latest approaches in the literature and classified the methodologies into six categories. In their analysis, they revealed that most of the common successful methodologies are based on meta-heuristics and hybrid

approaches. However, they acknowledged the difficulty of objectively comparing these approaches in their current form due to different problem formulations. Identifying the ITC 2019 as the most recent UCTP benchmark and the closest replica to real-world applications, they suggested that current methods be tested on this dataset. Additionally, they emphasized the importance of generating robust and flexible techniques with high adaptability, highlighting the limited generalizability of existing SOTAs as a barrier to their widespread adoption in real-world applications.

Recognizing the prominent role of (hybrid) meta-heuristics in the ETP domain, Abdipoor et al. [8] provided an exhaustive and detailed overview of these approaches for the UCTP. First, they presented a comprehensive overview of the problem, its variants, and benchmark datasets. Then, they conducted an extensive literature review, focusing on meta-heuristic approaches. In addition, they introduced a categorization of hybrid meta-heuristics based on their hybridization type into collaborative and integrative. They analyzed the strengths and limitations of different approaches and explored possible future directions. They highlighted the gap between real-world UCTP and the literature, resulting in many universities relying on manual timetabling due to the lack of generality in current methods and excessive parameter tuning. They advocated for enhancing generality and its assessment to bridge this gap and discourage problem-specific solutions. Finally, they pointed out the untapped potential of hybrid meta-heuristics for the ITC 2019, despite their proven effectiveness as SOTA on previous UCTP benchmarks.

As one of the few works solely focusing on the ITC 2019, [17] delivered a systematic literature review of this latest UCTP benchmark dataset. In this brief survey, they first described the problem definition, its constraints, and performance measurement. Then, they reviewed and classified the existing methods in the literature based on their encoding, preprocessing, and optimization steps. Having identified the competition winners and current SOTAs based on detailed empirical results, they pinpointed the complications in conducting a fair comparison between different approaches as the foremost obstacle faced in this field. They noted that the current methods vary in their implementation (using diverse programming languages, encodings, and preprocessing steps), testing conditions (different time limits and test benches, sometimes using unrealistically powerful servers), and reporting standards (lacking exact parameters, source code, and preprocessed problem instances). They also observed a lack of generality in the methodologies and attributed the scarcity of work on this benchmark to its novelty and complexity. They referred to an open-source C# implementation[1] of ITC 2019 and shared their own Java implementation[2] GitHub repository to encourage further research on this problem.

[1]https://github.com/edongashi/itc-2019 Last accessed: 22 December 2024

[2]https://github.com/SinaAbdipoor/itc-2019 Last accessed: 22 December 2024

Despite the growing recognition of method generality as a crucial aspect of OR in the literature, there is still a lack of empirical studies that exploit and apply this concept in practice. Pillay et al. [18] examined the generality performance of hyper-heuristics in discrete optimization problems. They argued that the existing evaluation of hyper-heuristics mainly focuses on optimality, which overlooks their potential as general solvers. To address this gap, they proposed a formal definition of generality and suggested four levels of generality for hyper-heuristics. They also developed two metrics for assessing the generality performance of these approaches and demonstrated their usefulness through several case studies. They revealed how measuring generality can help avoid overfitting an algorithm to a few instances and ensure its robustness across different instances. They concluded that generality performance assessment is essential for providing a complete understanding of a method's performance. They highlighted the importance of considering both generality and optimality for a fair comparison of hyper-heuristics and encouraged more research on this topic.

While the number of studies on the concept and implications of generality remains limited, especially on meta-heuristics, there have been several studies on introducing hyper-heuristic-based approaches with high generality. Misir et al. [43] investigated the generality of selection hyper-heuristics by examining their performance across different problems. They reported that while the term generality is often used in the hyper-heuristic literature, the actual generality performance analysis is often overlooked. Moreover, following their experimental study, they revealed significant performance variations under diverse conditions in their studied hyper-heuristics, further emphasizing the need for generality research as a future direction. In another research work [33], they introduced a new adaptive selection hyper-heuristic, designed for enhanced generality, which demonstrated effective performance across six problem domains in the HyFlex framework. Finally, Park et al. [44] investigated the generality of genetic programming-based hyper-heuristics for dynamic job shop scheduling problems. They developed a specialized dataset and analytical method to assess the method's adaptability when dealing with machine breakdowns.

IV. THEORETICAL ANALYSIS

This section delivers the theoretical foundations on which this research is based. It presents the theoretical analysis behind the optimality vs. generality trade-off as the primary inspiration of this work. To demonstrate how optimizing a method's performance on a single problem instance by parameter tuning can negatively affect its performance consistency over other problem instances, we use a well-established theorem and mathematical equations that explain the relationship between optimality and generality.

A. NO FREE LUNCH THEOREM

The No Free Lunch (NFL) theorem states that when averaged over the entire space of discrete functions, all

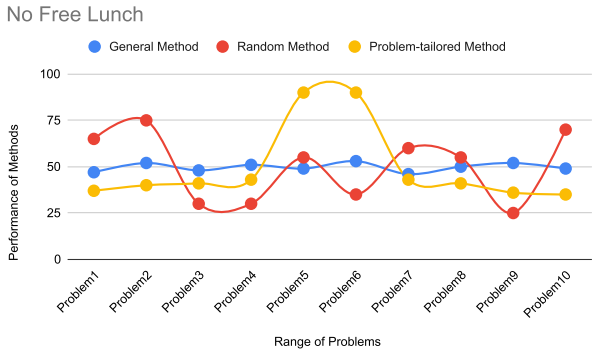


FIGURE 3. A conceptual visualization of the No Free Lunch Theorem as discussed in [45]. This illustration conveys the principle that no optimization algorithm is universally superior across all problems.

search algorithms exhibit equivalent performance across all possible performance measures [12]. In other words, the expected performance of different algorithms averaged over the entire function space is the same [45]. From this, it can be interpreted that the superior performance of a method, $method_1$, on a specific problem, $problem_x$, compared to another method, $method_2$, comes at the cost of $method_1$ attaining a lower performance on another problem, let's say $problem_y$, compared to $method_2$. For instance, a randomly selected method from the space of all methods, a problem-tailored method with high optimality on specific problem instances, and a general method designed for consistent performance without extensive problem tuning would theoretically all perform similarly on average across all problems. This concept is roughly depicted in Figure 3 and can be mathematically represented as follows, where $U(F)$ is the uniform distribution over the set of all possible objective functions, $\mathbb{E}$ denotes the expectation operator, and $x \sim a(f)$ and $x \sim b(f)$ mean that x is sampled from the distributions induced by algorithms a and b on function f.

$$E_{f\sim U(F)}[\mathbb{E}_{x\sim a(f)}[f(x)]] = \mathbb{E}_{f\sim U(F)}[\mathbb{E}_{x\sim b(f)}[f(x)]]$$

NFL was first explored and proved by Wolpert et al. [46] and was later expanded in [47]. It is still an active field of research, stimulating the curiosity of field experts and evoking challenging yet fascinating philosophical debates. Some recent works in the NFL literature include [48], [49], and [50].

B. OPTIMALITY VS. GENERALITY TRADE-OFF

An intriguing extension of the NFL and its equation involves considering two configurations of the same method, denoted $a(f)$ and $b(f)$. According to the NFL theorem, these two algorithms will have equal average performance over all possible problems. Hence, it can be deduced that tuning the parameters of an algorithm to enhance its effectiveness on a specific problem may adversely affect its effectiveness on other problem instances. This highlights the inherent trade-off between optimality and generality, emphasizing the critical role of parameter tuning. This trade-off in the context of meta-heuristics has been extensively explored and theoretically investigated in various studies.

The computational time complexity of Evolutionary Algorithms (EA) was examined by He and Yao [51], in which they utilized drift analysis to estimate the average time complexity order of an EA when applied to a problem based on the upper and lower bounds. To extend this, they [52] performed a time complexity analysis on both $(1+1)$ and $(N+N)$ EAs by deriving their First Hitting Time (FHT) on a variety of problems. They directly estimated the FHT of a Markov chain from the transition matrix. Through further empirical studies, they concluded that increasing the population size, one of the key characteristics of EAs, can lower time complexity, boost first-hitting probabilities, and shorten FHT. Nonetheless, they emphasized their results do not imply that $(N+N)$ will always outperform $(1+1)$ on all possible problems. Given the distribution μ_0 of initial population X_0 conditional on initial state i with the transition probability $\mathbb{P}$, they formulated $\mathbb{E}[T]$ as the mean first hitting time T as follows:

$$\mathbb{E}[T] = \sum_i \mu_0(i) \sum_{t=0}^{\infty} t\mathbb{P}(T = t \mid X_0 = i)$$

The significance and importance of parameter tuning on the performance of meta-heuristics was reiterated by Yu et al. [53]. Identifying the population size as the main parameter of Genetic Algorithms (GA), they introduced a population-sizing model for the entropy-based model building. Based on entropy-based modeling and selection pressure, they proposed that $n = \Theta(m \log m)$ is the optimal population size based on the following three facetwise models:

1) **Building-Block Supply Model:** given the alphabet cardinality (χ), the Building Block (BB) size (k), and the number of BBs (m), the required population size (n) with a tolerance of $\epsilon = \frac{1}{m}$ is calculated from [54]:
$$n = \chi^k(k \log \chi + \log m)$$
2) **Gambler's Ruin Population-Sizing Model:** given the distance between competing BBs (d) and the fitness variance of a building block (σ_{BB}), the approximate population size with a failure probability of $a = \frac{1}{m}$ is given by [55]:
$$n = \frac{\sqrt{\pi}}{2}\frac{\sigma_{BB}}{d}2^k\sqrt{m}\log m$$
3) **Model-Building+Decision Making Population Sizing:** the ideal population size according to this model is as follows, where c_n is a problem-dependent constant and the failure rate is proportionate to $a = \frac{1}{m}$ [56]:
$$n = c_n.2^k(\frac{\sigma_{BB}}{d})^2 m \log m$$

In addition to the aforementioned theoretical investigations, there have been numerous experimentations attesting to these observations. Chen et al. [57] revealed that increasing

population size can be detrimental to the performance of EAs. Minnaert et al. [58] raised the interesting question of "to tune or not to tune?" and displayed the inescapable trade-off in the pursuit of optimality. Similarly, Weise et al. [59] challenged the common myth that "An EA will outperform a local search algorithm, given enough runtime and a large enough population" and showed this might not always hold true. Finally, Roeva et al. [60] explored the optimal parameter tuning of Genetic and Ant algorithms in a specific problem.

The trade-off between optimality and generality in parameter tuning poses a fundamental dilemma: is tuning worth it? We argue that the choice of an approach or the setting of its parameters depends on the problem and the algorithm used. Direct solvers and heuristics are effective problem-tailored solutions and should be tuned to fit the problem instance, while hyper-heuristics are general solvers and should be tuned for higher generality levels. Correspondingly, for a benchmark dataset such as ITC 2019, a proposed meta-heuristic should be able to perform relatively consistently across all instances rather than being tuned for a few specific instances.

V. PROPOSED FRAMEWORK

This section introduces our measurement framework, considering both optimality and generality for a comprehensive evaluation of method effectiveness in our experimental study.

A. PERFORMANCE EVALUATION

In order to define a performance assessment framework, this section presents evaluation metrics for optimality and level 1 generality measurements of meta-heuristics in ETP. We utilize the ITC 2019 as the benchmark. However, this framework can be applied to other ETPs and extended to adopt other problems and higher generality levels for hyper-heuristics.

1) GENERALITY LEVEL 0 (OPTIMALITY)

The optimality measurement for a specific problem instance is inherently dictated by how well it performs on that specific instance, which, in the case of ETPs, is guided by feasibility and the fitness function.

1) **Feasibility:** a Candidate Solution (CS) is feasible if and only if it satisfies all the Hard Constraints (HC) of that Problem instance (P_i). An infeasible solution is generally deemed worthless in the literature [1], [8].

$$feasible_{P_i}(CS) = \begin{cases} true & \iff CS \text{ satisfies all } HC \\ false, & \text{Otherwise} \end{cases} \tag{1}$$

2) **Cost:** given all classes $\{c_1, c_2, c_3, \ldots, c_i\}$, all distribution constraints $\{d_1, d_2, d_3, \ldots, d_j\}$, and all students $\{s_1, s_2, s_3, \ldots, s_k\}$, the total Cost ($C$) of a candidate solution for a given problem instance is calculated as follows [17].

$$C_{P_i}(CS) = W_t \sum_{a=1}^{i} T_{c_a} + W_r \sum_{a=1}^{i} R_{c_a} + W_d \sum_{b=1}^{j} D_b + W_s \sum_{c=1}^{k} SC_{s_c} \tag{2}$$

where T_{c_a} is the penalty of the Time assigned to class a, R_{c_a} is the penalty of the Room assigned to class a, D_b is the penalty of the Distribution constraint b, SC_{s_c} is the Student Conflict penalty of student c, and W_t, W_r, W_d, W_s are the predefined weights of these penalties, respectively. In simpler terms, the total cost is a weighted sum of the time, room, distribution constraint, and student conflict penalties. These weights are specific to each problem instance and are defined as part of the instance attributes [61].

The cost function (2) is applicable to a wide range of ETPs. In specific problems that do not require certain penalty aspects, the corresponding weights can be set to zero. For instance, to adapt this cost function to an ETP problem that does not involve the student sectioning problem, such as ITC 2002, we can set W_s to zero.

Two primary challenges hinder the adoption of this cost function as the fitness function in measuring optimality.

1) In many cases, zero-cost solutions are either nonexistent or are yet to be identified, resulting in inflated cost calculations. To address this issue, we reformulate the cost function to assess cost relative to the current best solution rather than absolute zero.

$$C^1_{P_i}(CS) = C_{P_i}(CS) - C_{P_i}(BKS)$$

2) Furthermore, the cost function values are often diverse, with values for larger problem instances typically much higher, which leads to a bias towards larger problem instances. To address this issue, we further modify the cost function by scaling and normalizing its values between 0 and 1. The addition of the constant (1) to the denominator prevents division by zero.

$$C^2_{P_i}(CS) = \frac{C_{P_i}(CS) - C_{P_i}(BKS)}{C_{P_i}(CS) + 1}$$

Given Eq. (1) and Eq. (2), we define the Optimality of a given Candidate Solution for a Problem instance ($O_{P_i}(CS)$) using the feasibility and fitness metrics as follows:

$$O_{P_i}(CS) = \begin{cases} feasible_{P_i}(CS), \\ \quad \text{Refer to Eq. (1)} \\ fitness_{P_i}(CS) = 1 - \dfrac{C_{P_i}(CS) - C_{P_i}(BKS)}{C_{P_i}(CS) + 1}, \\ \quad \text{Refer to Eq. (2)} \end{cases} \tag{3}$$

Feasibility determines whether the candidate solution adheres to all hard constraints of the problem instance.

Fitness, represented by a value between 0 and 1 (higher values indicating better quality), measures how closely the candidate solution's quality aligns with the best-known result in terms of soft constraint violations.

The average of $(O_{P_i}(CS))$ over all instances provides a comprehensive assessment of the solver's optimality on the entire dataset. Table 3 illustrates an example of optimality and level 1 generality measurements in a hypothetical scenario.

2) GENERALITY LEVEL 1

The level 1 generality function seeks to quantify the consistency of an algorithm's optimality performance across all problem instances in a given benchmark dataset. While this measure allows some problem adaptation, it penalizes problem-tailored solutions. Compared to the current optimality metrics, this approach offers significant advantages, particularly for meta-heuristics. It provides a more comprehensive understanding of a proposed method's capabilities by assessing performance over the entire dataset rather than focusing on a few select instances. This broader perspective simplifies comparisons and promotes closer alignment to real-world applications. Moreover, it discourages the pursuit of limited optimality as the sole criterion for publication in this field. Although research on generality has remained limited, several measurement techniques have been proposed.

Pillay et al. [18] explored the Ranking algorithm as a potential tool for evaluating generality performance. This method sorts a group of n solvers based on their performance, assigning ranks from 1 to n. These ranks are then aggregated across all problem instances to determine the overall generality. While Ranking provides a relative comparison by indicating which solver produces the best objective values, it overlooks performance consistency. For example, in Table 3, solver 1 exhibits marginal fitness superiority over solver 2 for problem instances 1 and 3, achieving gains of 0.2 and 0.3, respectively. However, its performance drops significantly on problem instance 2, with a 0.5 reduction in fitness. Despite Ranking favoring solver 1, this performance inconsistency contradicts the principles of generality. Additionally, this algorithm is only applicable when evaluating multiple solvers, limiting its usefulness. Therefore, it is not considered an ideal generality measurement technique.

Another rudimentary approach to evaluating generality performance involves directly calculating fluctuations by simply summing the differences in the optimality performance of solvers across all instances. While this method is extremely straightforward, it fails to distinguish between a series of minor deviations and a single significant deviation if they are equal. For instance, in Table 3, both solvers exhibit the same total added difference across all instances $((0.7-0.5)+(0.3-0.8)+(0.9-0.6))=0$. However, solver 1 is clearly less general due to its substantial performance drop on instance 2. This highlights the inadequacy of this approach, as an ideal generality measurement should penalize larger deviations more severely than smaller ones.

To overcome these challenges, we propose employing Standard Deviation (SD) as a measuring stick for performance generality. SD is a measure of the dispersion in data and is defined as the square root of the variance, which is the average of the Sum of Squared Differences (SSD) from the mean. In other words, SD tells us how far, on average, the data points are from the mean [62], making it an ideal tool for assessing generality. SD has been explored as an evaluation metric for performance assessment in hyper-heuristics by Pillay et al. [18] and was found to be highly effective in that context. We aim to extend its application to meta-heuristics in the context of ETPs. The mathematical formulation of SD is as follows, where σ is the standard deviation, x_i is the i-th data point, μ is the mean, and n is the number of data points.

$$\sigma = \sqrt{\frac{\sum(x_i-\mu)^2}{n}}$$

Level 1 generality of a given solver across all instances of a dataset ($G(solver)$) is measured by the SD of fitness values (refer to Eq. (3)) across all problem instances, given $\mu_{fitness}$ as the average of all fitness values, and is formulated as follows:

$$G(solver) = 1 - \sqrt{\frac{\sum_{i=1}^{n}(fitness_{P_i}(CS_i)-\mu_{fitness})^2}{n}} \quad (4)$$

Table 3 illustrates the computation of Level 1 generality. Higher generality indicates better performance consistency across different instances. Due to the stochastic nature of meta-heuristics [45], it is advisable to average the optimality and generality metrics of a solver over at least three independent, consecutive runs to mitigate the impact of randomness. Furthermore, for a more comprehensive comparison of the performance of different solvers, the t-test [63] can be employed, utilizing the averaged optimality and generality metrics.

3) HIGHER GENERALITY LEVELS

One of the key advantages of this assessment framework is its expandability. While this study focuses on meta-heuristics and level 1 generality in ETPs, the core concepts of optimality and generality can easily be extended to other problems and higher generality levels.

To apply this framework across different benchmark datasets for a single problem (level 2 generality), such as evaluating a meta-heuristic on ITC 2019, ITC 2007, and ITC 2002, the optimality value is first computed for each individual instance. Then, an average optimality value is calculated for each benchmark dataset. Level 2 generality is then determined from the SD of these average optimality values.

To extend this framework to different problems within a domain (level 3a generality), such as solving both university course timetabling and examination timetabling, the optimality value is first computed separately for each problem. For course timetabling, the optimality value is obtained by averaging the solver's performance across all relevant

TABLE 3. An artificial example demonstrating performance evaluation.

Problems	Solvers							
	solver 1				solver 2			
	feasible?	cost	BKS	fitness	feasible?	cost	BKS	fitness
instance 1	yes	1.86	1	0.7	yes	3	1	0.5
instance 2	no	215.67	64	0.3	yes	80.25	64	0.8
instance 3	yes	2.89	2.5	0.9	yes	4.83	2.5	0.6
Optimality	Avg(feasibility) 66.67%		Avg(fitness) 0.63		Avg(feasibility) 100%		Avg(fitness) 0.63	
Level 1 Generality	0.69				0.85			

benchmark datasets. The same is done for the examination timetabling. Level 3a generality is then calculated from the SD between these two resulting optimality values.

For higher generality levels (3b and 4), which primarily fall within the scope of hyper-heuristics, the same methodology applies. Optimality values are extended across broader problem domains, and generality levels are determined by computing the SD of optimality values from lower levels.

4) OTHER MEASUREMENTS

Alongside the previously mentioned optimality and generality metrics, the following additional measurements can provide a deeper insight into a solver's performance, facilitating a more objective comparison of algorithms and aiding researchers in conducting more rigorous studies.

- **Running time** is an essential method's performance measurement in optimization problems, defining the execution time limit for algorithms. While often considered an objective in COPs, running time is highly dependent on the specifications of the testbench used, making direct comparisons and experiment replication potentially misleading. As a positive initiative, the ITC 2007 dataset came with benchmarking software to measure host computer speed and assign a time limit to mitigate the biased effects of superior hardware on performance comparisons. Unfortunately, this practice was not continued in subsequent datasets.
- **Number of Fitness Evaluations** or NFE [64] is a valuable alternative to running time for measuring the performance of meta-heuristic algorithms. While using benchmarking scores to determine time limits can help mitigate hardware bias and make the running time a more viable performance metric, it remains sensitive to programming language, implementation efficiency, and other runtime factors that can obscure the true performance of the solver. In meta-heuristics, the most computationally intensive step is the fitness evaluation of newly generated solutions [65], making NFE an ideal performance measure that eliminates external influences and reveals the algorithm's true performance. Maximum NFE can serve as a termination criterion and time budget for algorithm execution, enabling fair comparisons of meta-heuristic performance. However, NFE's efficacy is primarily limited to meta-heuristics. While it can be applied to other approaches, its accuracy may be compromised, as solution creation can be more computationally demanding in other methods, and fitness evaluation may not be the sole significant step. While the utilization of NFE has been a common practice in other meta-heuristic applications [66], [67], [68], [69], its potential in ETPs has been overlooked. We have incorporated NFE into our performance assessment framework.
- **Hardware specifications**, including CPU specifications, RAM specifications, GPU specifications (if applicable), storage speed, operating systems, and benchmarking scores, can provide valuable context for future research, where NFE is not applicable. This information can help researchers understand and account for potential performance variations due to hardware differences.

B. PERFORMANCE ASSESSMENT FRAMEWORK

Considering the performance measurements introduced in Section V-A, Algorithm 1 presents the pseudocode for the proposed performance evaluation framework. This framework is intended as a standalone reference for future research on meta-heuristics in ETPs. Note that the evaluation of a candidate solution (in line 14) is conducted before calculating optimality (in line 20). In the case that this current solution is a new best solution, BKS is updated first before fitness is calculated. Therefore, in the case of a new BKS, the fitness value reaches its highest value of 1.

VI. EMPIRICAL STUDY

To observe the effectiveness of our proposed evaluation framework in practice, this section applies it to the best-performing approaches on the latest UCTP benchmark dataset from the literature, demonstrating its ability to facilitate fair performance comparisons.

A. BENCHMARK SETUP

This section introduces the ITC 2019 dataset, presents leading literature approaches, and outlines the utilized optimality and generality evaluation metrics for analysis.

1) BENCHMARK DATASET

The ITC 2019 is currently the latest benchmark dataset on the UCTP. It comprises 30 real-world problem instances. In addition to the Course Timetabling Problem present in

Algorithm 1 Meta-Heuristics Level 1 Performance Assessment Framework

```
procedure Optimality_VS_Generality(dataset, solver)
    initialize solver
    for i ← 0 to number_of_instances do
        problem_instance ← dataset[i]
        for r ← 0 to number_of_runs do
            procedure Algorithm(problem_instance)
                solution ← initialsolution
                timer ← 0
                NFE ← 0
                while termination condition is not met do
                    procedure run(solution)
                        resume timer
                        solution ← newsolution
                        evaluate solution        ▷ (refer to Eqs. (1), (2))
                        NFE ← NFE + 1
                        pause timer
                    end procedure
                end while
                save solution
                calculate Optimality  ▷ (refer to Eq. (3))
            end procedure
        end for
        report (average, best, worst) of optimality, NFE, timer in r runs
    end for
    calculate Generality from average(optimalities)  ▷ (refer to Eq. (4))
end procedure
```

previous UCTP datasets, ITC 2019 includes the Student Sectioning Problem. While this helps bring this benchmark closer to real-world scenarios, it significantly increases the complexity of this problem [17]. This dataset is explained in detail in [19] and [61]. All information about this dataset is available on the ITC 2019 official website.[3] Due to its recency and high implementation complications, the number of conducted works on this dataset has remained relatively limited. Achieving fair performance comparison on ITC 2019 is specifically problematic due to its challenging nature, its disproportionate problem size across instances, the wide range of approaches in the literature, and the dissimilar running conditions of such methods. This further highlights the need for and the significance of our performance framework.

2) APPROACHES

A diverse array of meta-heuristics and matheuristics have been applied to the ITC 2019 during the competition and post-competition in the literature.

Holm et al. [70] proposed a multi-stage graph-based matheuristic employing Mixed-Integer Programming (MIP) and a local search to address the ITC 2019. Their utilized preprocessing method was further explained in [71]. This method won the competition with 489 points, and its extended version [72] is the current SOTA. They [73] later introduced further data reduction improvements. The DSUM team actively contributes to the ITC 2019 and shares their latest advancements and reduced datasets on their website.[4] to promote further research.

Another matheuristic MIP model was introduced by Rappos et al. [74]. Achieving the second rank in ITC 2019 with 322 points, the first stage of this approach deals with finding feasible solutions, while the second stage employs a local search to reduce soft constraint violations. An extended version of this work was later published in [75].

A Simulated Annealing (SA) technique was proposed by Gashi et al. [76]. In the first stage of this multi-stage meta-heuristic approach, a feasible solution is found. In the second stage, the quality of this solution is enhanced while maintaining feasibility. Incorporating a gradual penalization procedure to escape local optima, this method attained 273 points and ranked third in the ITC 2019. They also won the best open-source award. A more comprehensive explanation of their approach was published in [77].

Er-Rhaimini [78] presented a multi-stage meta-heuristic based on Forest Growth Optimization to address the ITC 2019. Despite its straightforward procedure, this approach managed to attain 252 points, securing fourth place in the competition.

Finally, Lemos et al. [79] employed a Maximum Satisfiability (MaxSat) and a local search in a multi-stage matheuristic approach to address the ITC 2019. While MaxSat strived to find a feasible solution in the first stage, the local search procedure improved the quality of the solution in the second stage. Achieving 79 points in the fifth place, an extended version of this work was published in [80]. The source code for this method is available in a GitHub repository.[5]

Table 4 summarizes the literature-based ITC 2019 methods employed in our empirical study.

3) MEASUREMENT

Guided by the evaluation framework outlined in Section V-A, we employ the following performance metrics in our experimental analysis.

- Optimality (refer to Eq. (3)) of a candidate solution on each problem instance is jointly determined by its fitness and feasibility (refer to Eq. (1)). The total cost governs the fitness value (refer to Eq. (2)), benchmarked against the latest BKS cost from the ITC 2019 website.[6]

[3] https://www.itc2019.org/home Last accessed: 22 December 2024

[4] https://dsumsoftware.com/itc2019/ Last accessed: 3 September 2024

[5] https://github.com/ADDALemos/MPPTimetables Last accessed: 22 December 2024

[6] https://www.itc2019.org/results Last accessed: 24 April 2024

TABLE 4. Summary of the ITC 2019 competition methods in the literature.

Author	Ranking	Points	Method	Category	Single/Multi Stage	Max Runtime
Mikkelsen et al. [72]	1st	489	Parallelized Matheuristic	Matheuristic	Multi-stage	627 hours
Rappos et al. [75]	2nd	322	Mixed Integer Programming	Matheuristic	Multi-stage	178 hours
Sylejmani et al. [77]	3rd	273	Simulated Annealing	Meta-heuristic	Multi-stage	24 hours
Er-rhaimini [78]	4th	252	Forest Growth	Meta-heuristic	Multi-stage	50 hours
Lemos et al. [80]	5th	79	MaxSat	Matheuristic	Multi-stage	6000 seconds

TABLE 5. Optimality analysis of the existing literature for solving the ITC 2019 benchmark.

	Approach		Mikkelsen et al. [72]			Rappos et al. [75]			Sylejmani et al. [77]			Er-rhaimini [78]			Lemos et al. [80]		
#	Name	BKS	Feasible	Fitness	Cost	Feasible	Fitness	Cost	Feasible	Fitness	Cost	Feasible	Fitness	Cost	Feasible	Fitness	Cost
1	agh-fis-spr17	2985	1	0.9688513952	3081	1	0.6551118912	4557	1	0.4391176471	6799	1	0.5229422067	5709	1	0.08497438816	35139
2	agh-ggis-spr17	34285	1	0.9574687928	35808	1	0.9363410438	36616	1	0.4399420015	77932	1	0.6040947213	56755	1	0.2127992353	161118
3	bet-fal17	289452	1	0.9978144488	290086	1	0.9797751059	295427	1	0.9674037285	299205	1	0.9223741528	313812	1	0.9778289011	296015
4	iku-fal17	18968	1	1	18968	1	0.7067173354	26840	1	0.3747777295	50613	1	0.4264325697	44482	1	0.6337788172	29929
5	mary-spr17	14910	1	1	14910	1	0.9926108374	15021	1	0.9380937402	15894	1	0.8929277202	16698	1	0.2915265504	51147
6	muni-fi-spr16	3752	1	0.9989353207	3756	1	0.9760728218	3844	1	0.7495506291	5006	1	0.7206221198	5207	1	0.1943049443	19314
7	muni-fsps-spr17	868	1	1	868	1	0.9830316742	883	1	0.4481691594	1938	1	0.210106383	4135	1	0.004115694103	211142
8	muni-pdf-spr16c	32762	1	0.8979116422	36487	1	0.8739596671	37487	1	0.5628704451	58206	1	0.4223451156	77573	1	0.05769139339	567900
9	pu-llr-spr17	10038	1	1	10038	1	0.7499626475	13385	1	0.5949037037	16874	1	0.5219945923	19231	1	0.1476236692	68003
10	tg-fal17	4215	1	1	4215	1	1	4215	1	0.5240522063	8044	1	0.5729039272	7358	1	0.6222878229	6774
11	agh-ggos-spr17	2855	1	0.9345549738	3055	1	0.4518272425	6320	1	0.3061421374	9328	1	0.3696608853	7725	1	0.03581370852	79745
12	agh-h-spr17	21161	1	0.9003956942	23502	1	0.8089449541	26159	1	0.8437126226	25081	1	0.8219529247	25745	1	0.3786501575	55887
13	lums-spr18	95	1	1	95	1	0.8347826087	114	1	0.8888888889	107	1	0.5363128492	178	1	0.8	119
14	muni-fi-spr17	3738	1	0.9772608468	3825	1	0.8715617716	4289	1	0.7967185169	4692	1	0.6880750828	5433	1	0.2067916598	18080
15	muni-fsps-spr17c	2594	1	0.9992298806	2596	1	0.7854116223	3303	1	0.2813618129	9222	1	0.1103269419	23520	1	0.004197548438	618217
16	muni-pdf-spr16	17159	1	0.9453503746	18151	1	0.7056211193	24318	1	0.4281971304	40074	1	0.4419604914	38826	1	0.05517773598	310994
17	nbi-spr18	18014	1	1	18014	1	0.9453715365	19055	1	0.6793498756	26517	1	0.5943582976	30309	1	0.3608412619	49924
18	pu-d5-spr17	15184	1	0.9543711897	15910	1	0.8071117253	18813	1	0.7810812201	19440	1	0.7501358494	20242	1	0.8670206692	17513
19	pu-proj-fal19	117169	1	0.7915982624	148016	1	0.2087866072	561194	1	0.4924971628	237909	1	0.6655873665	176039	1	0.9257401101	126568
20	yach-fal17	1074	1	0.8669354839	1239	1	0.5826558266	1844	1	0.6221064815	1727	1	0.3378378378	3181	1	0.03338613	32198
21	agh-fal17	117627	1	0.6317259306	186200	0	0		1	0.6391749216	184030	1	0.7676213969	153236	1	0.8243720565	142687
22	bet-spr18	348524	1	0.9998135345	348589	1	0.9679690494	360057	1	0.9669485459	360437	1	0.9342831868	373039	1	0.9847536597	353920
23	iku-spr18	25863	1	0.9994203795	25878	1	0.7045107867	36711	1	0.3008491334	85969	1	0.3646257736	70932	1	0.5679652159	45537
24	lums-fal17	349	1	1	349	1	0.9043927649	386	1	0.7186858316	486	1	0.626118068	558	1	0.42997543	813
25	mary-fal18	4331	1	0.97920434	4423	1	0.7683575736	5637	1	0.6016666667	7199	1	0.6237580994	6944	1	0.09823574765	44097
26	muni-fi-fal17	2837	1	0.946	2999	1	0.747826087	3794	1	0.6021642266	4712	1	0.5886745488	4820	1	0.1441780126	19683
27	muni-fspsx-fal17	10014	1	0.5865300146	17074	1	0.3034664566	33001	1	0.2388276816	41933	1	0.09572190469	104625	1	0.02496535014	401155
28	muni-pdfx-fal17	82258	1	0.7005953344	117412	1	0.5430891625	151464	1	0.5166892792	159203	1	0.4286823564	191887	1	0.3598995454	228560
29	pu-d9-fal19	38834	1	0.9029925361	43006	1	0.2897918066	134009	1	0.4692597694	82757	1	0.5512341911	70450	1	0.5400951268	71903
30	tg-spr18	12704	1	1	12704	1	0.9881776464	12856	1	0.7944100544	15992	1	0.6436496276	19738	1	0.3982633773	31900

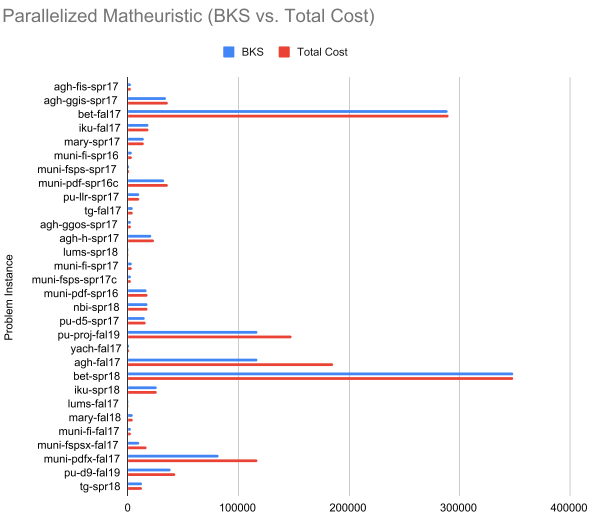


FIGURE 4. Total cost as reported in [72] vs. best known solution cost. It is evident that certain problem instances have significantly higher cost values.

To ensure accuracy, we evaluate both total cost and feasibility using the ITC 2019 online validator.[7]

- Level 1 Generality (refer to Eq. (4)) of a proposed method over the entire dataset is determined by the average of the method's optimality performance values on individual problem instances, along with an assessment of the deviation in fitness values.

[7]https://www.itc2019.org/validator Last accessed: 21 September 2024

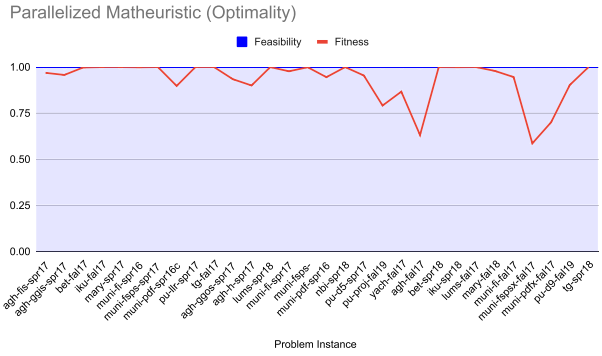


FIGURE 5. Optimality performance evaluation of the solver proposed in [72].

B. EMPIRICAL RESULTS AND DISCUSSION

This section showcases the usefulness of our performance evaluation framework in action by applying it to literature methods. We demonstrate how it can pave the path for fair performance comparison of different methods and provide further insights beyond current evaluation metrics.

Table 5 provides a detailed overview of the optimality performance of the studied literature methods across all

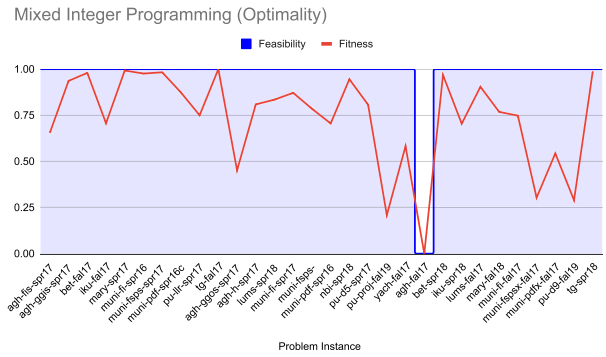


FIGURE 6. Optimality performance evaluation of the solver proposed in [75].

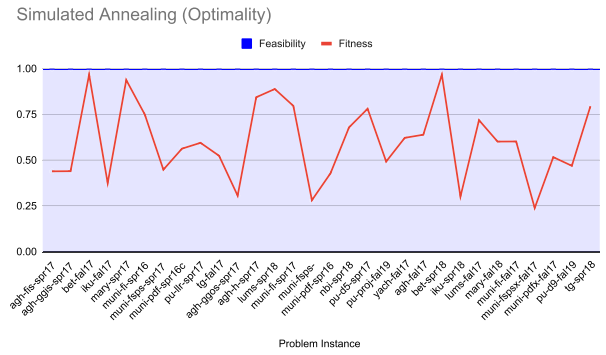


FIGURE 7. Optimality performance evaluation of the solver proposed in [77].

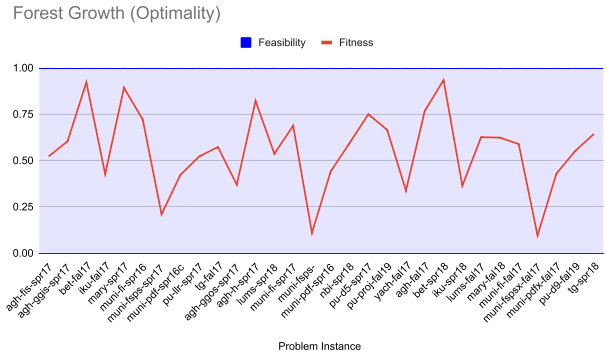


FIGURE 8. Optimality performance evaluation of the solver proposed in [78].

problem instances in terms of feasibility, total cost, and fitness. Note that the fitness values, given by Eq. (3), are between 0 and 1 and indicate how close a candidate solution's cost is to the current BKS.

The comparison of cost values produced by the current SOTA method [72] against BKS is depicted in Figure 4. It can be seen that for certain problem instances (particularly larger ones), both the total cost and BKS are exponentially higher than those in smaller instances. Consequently, there is

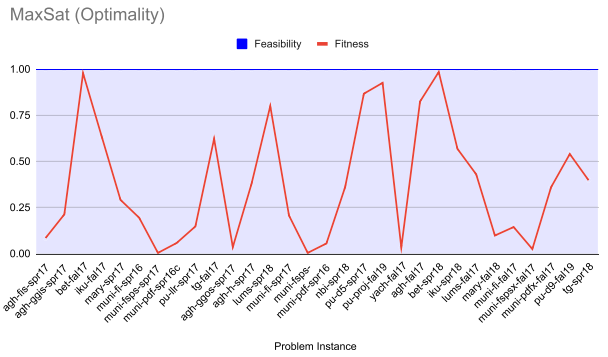


FIGURE 9. Optimality performance evaluation of the solver proposed in [80].

a greater potential for significantly reducing costs in larger problems compared to smaller ones. This creates a bias toward tuning a solver for larger problems to achieve a substantial overall cost reduction. The considerable variability in values across the ITC 2019 problem instances underscores the importance of a normalized metric to mitigate the bias toward larger instances.

Figures 5 to 9, drawn from Table 5, illustrate the optimality performance of methods by [72], [75], [77], [78], and [80], respectively. Significant fitness variation over instances is visible across the board. In particular, the MIP approach [75] did not produce a feasible solution in one instance. While constraint violations (both soft and hard) contribute to the total cost (2) and subsequently lower the fitness value, this approach still achieves a high average fitness compared to other methods that ensure full feasibility. This underscores a major limitation of using total cost as the sole evaluation metric, as feasibility is crucial in both practical and theoretical applications. To address this, our proposed optimality metric (3) reports both average fitness and feasibility rates for each solver, which are also incorporated into the framework 1.

Figure 10 provides an overview of the optimality performance comparison of literature methods across all problem instances of the ITC 2019. It can be seen that each method is capable of outperforming some of the other methods on one or a few instances at the cost of lower optimality performance over the rest of the instances. This confirms our theoretical observations (Section IV) and the limitations of existing metrics. Considering the existing performance metrics, these few optimality supremacies serve as sufficient indicators of the effectiveness of the proposed method, establishing a compelling case for publication. However, this encourages the pursuit of optimality via extreme parameter tuning and designing problem-tailored solutions. This approach sacrifices the overall generality of the proposed method and further widens the gap between literature and real-world applications. This gap is the main motivation behind this work, as we firmly believe the performance of a method should be assessed both in terms of optimality and

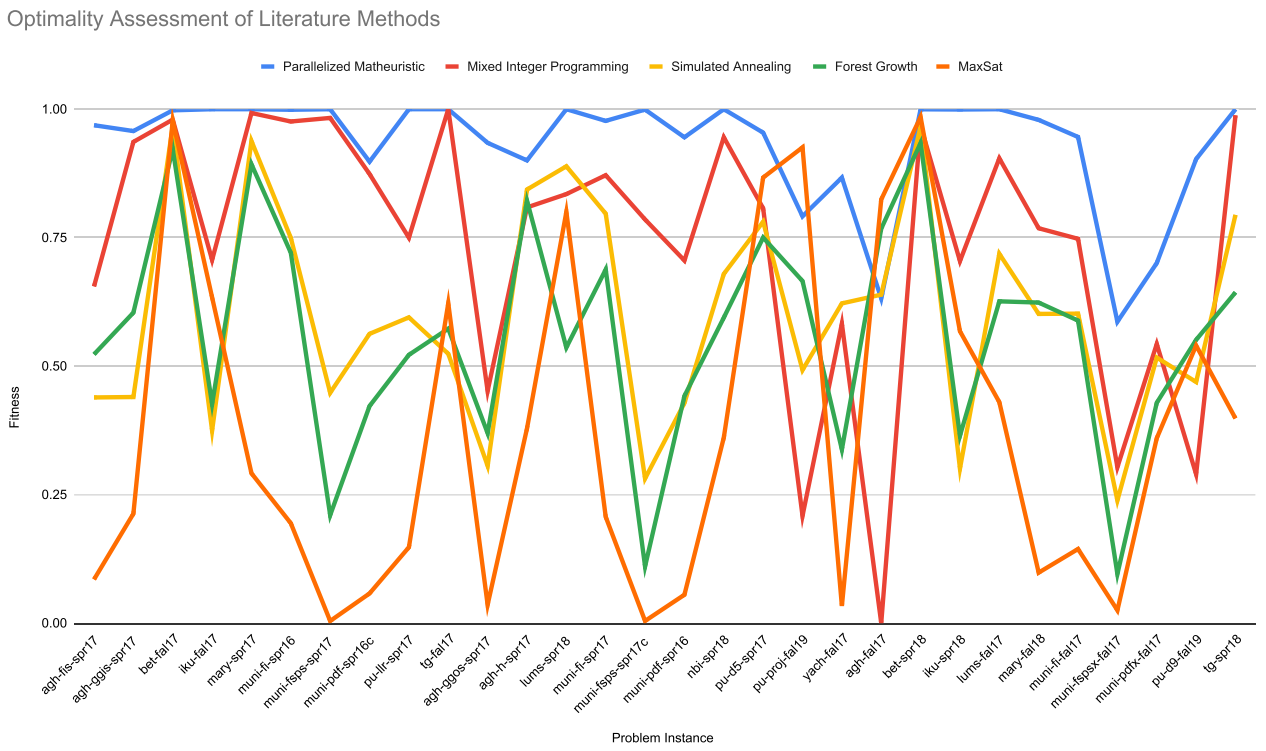


FIGURE 10. Comparative optimality analysis of the ITC 2019 literature. The high volatility of performance across the problem instances is evident.

TABLE 6. Comparison of level 1 generality metrics of literature methods.

Author	Feasibility Rate	Average Optimality	Level 1 Generality
Mikkelsen et al. [72]	1	0.9312320125	0.8883896246
Rappos et al. [75]	0.9666666667	0.7357746457	0.7399359611
Sylejmani et al. [77]	1	0.600253765	0.7893985769
Er-rhaimini [78]	1	0.558577373	0.7851766042
Lemos et al. [80]	1	0.3755751307	0.675592179

generality to align the literature more closely with real-world requirements.

Given the optimality performance of the methods, the level 1 generality evaluation metrics of these methods over the entire ITC 2019 dataset are summarized in Table 6.

Figure 11 analyzes level 1 generality, gathered from Table 6, of studied methods, revealing how our normalized optimality and generality metrics paint a clearer picture of their true performance. It also assists future research by fostering fair comparisons, regardless of the problem size.

Finally, Figure 12 encapsulates the essence of our proposed performance evaluation framework, depicting the relationship between optimality and generality performance across our studied methods. It is evident that the Parallelized Matheuristic method by Mikkelsen et al. [72] is the current SOTA, outperforming other methods both in terms of optimality and generality, while the MaxSat method proposed by Lemos et al. [80] seems to be tracking behind the other 4 methods in both metrics.

Optimality vs. generality performance comparison between the MIP approach by Rappos et al. [74] and the SA method by Sylejmani et al. [77] in Figure 12 brings out an interesting attribute. While the MIP has a higher average optimality performance (roughly 0.74 vs. 0.60), it has a lower level 1 generality (roughly 0.74 vs. 0.79) compared to the SA (refer to Table 6). As the current performance evaluation metric is solely based on optimality, the MIP is considered to be better performing and securing a higher rank in the ITC 2019 competition (refer to Table 4). However, Pareto optimality [81] paints a different picture, placing them in the same tier, distinct from the parallelized matheuristic, the forest growth, and the MaxSat in tiers 1, 3, and 4, respectively.

A compelling question raised when witnessing the MIP's dominance over other methods, both in terms of optimality and generality (Figure 12), is whether this goes against

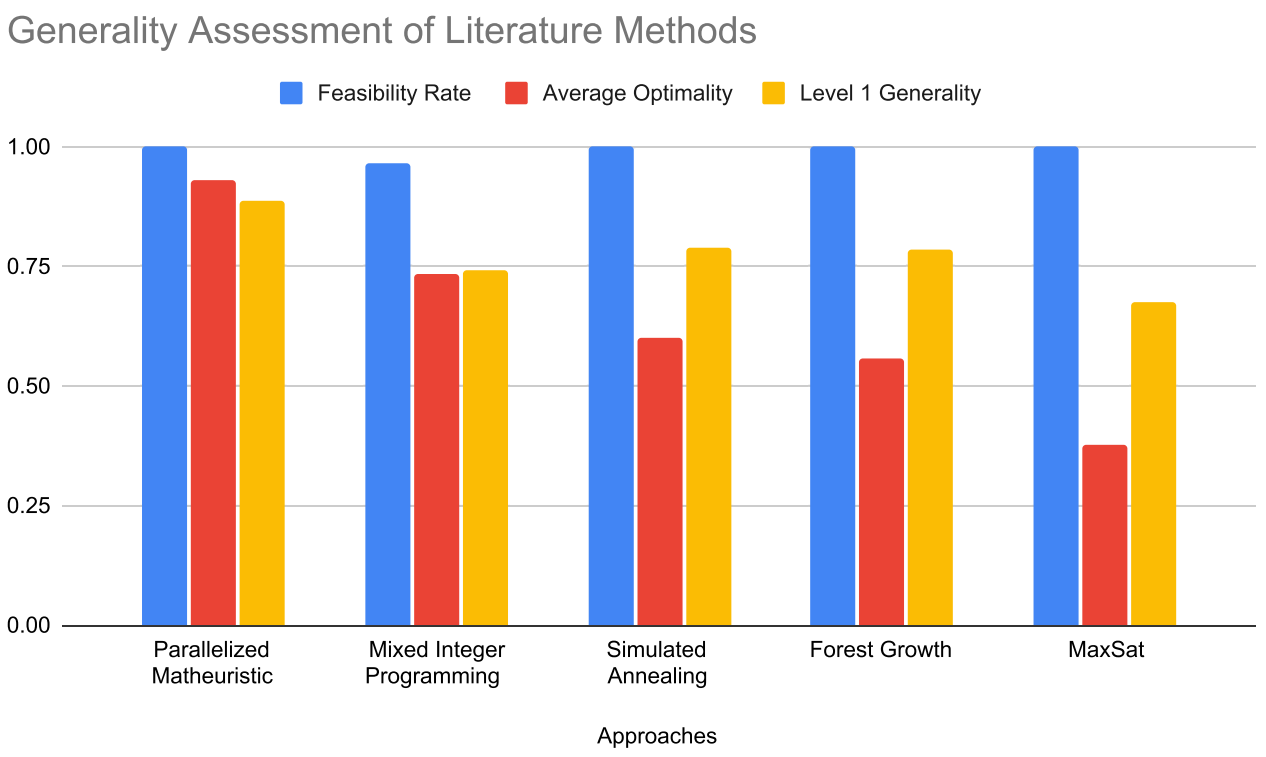


FIGURE 11. Comparison of level 1 generality performance of literature methods.

the claim of the NFL theorem (Section IV-A). We argue that this is not the case, and this performance dominance suggests a trade-off of performance in broader generality levels and other problems. However, this is a valid and intriguing question worth much more attention and research in the literature.

While our optimality and level 1 generality metrics offer a significant leap beyond the limitations of the current total cost metric, they are insufficient to ensure fair comparisons alone. In our study, the methods faced widely different experimental settings, from standard laptops [78] to powerful servers [72], and time limits ranging from a short 6000 seconds [79] to nearly a month [72] (see Table 4). In their work, Lemos et al. [80] showed that under similar running conditions, their method is far more competitive. Therefore, observing optimality vs. generality under uniform conditions becomes crucial for insightful comparisons. Hence, our proposed evaluation framework (Algorithm 1) takes the running conditions into account, paving the way for more impartial comparisons of meta-heuristics in future research.

VII. CONCLUSION AND FUTURE WORK

The low generalizability of current approaches in ETPs is a major challenge contributing to the existing gap between the literature and real-world applications and resulting in the wastage of resources. This study investigates this gap by analyzing the performance of current state-of-the-art approaches on the latest university course timetabling benchmark dataset, the International Timetabling Competition 2019. Following theoretical analyses, evaluation metrics are developed for the assessment of optimality and level 1 generality performance. Applying these metrics to current methods from the literature, we demonstrate this lack of generality and provide a comprehensive picture of their performance across both dimensions. Our investigation underscores the inadequacy of current evaluation metrics and their potential to favor problem-tailored approaches. To mitigate this, we incorporate penalization mechanisms in our framework to encourage more general approaches. Additionally, we present standard optimality and level 1 generality metrics, alongside other performance metrics, in our framework. Our framework establishes a fair and comprehensive basis for comparing solver performance in terms of both optimality and generality. We firmly believe our proposed framework introduces a new perspective to performance comparisons and opens avenues for interesting future research.

The optimality vs. generality performance assessment framework offers a more comprehensive overview of methods' performance in contrast to existing evaluation metrics. It is capable of closing the gap between literature and the real-world applications of meta-heuristics in ETPs by penalizing extreme parameter tuning/problem-tailored solutions

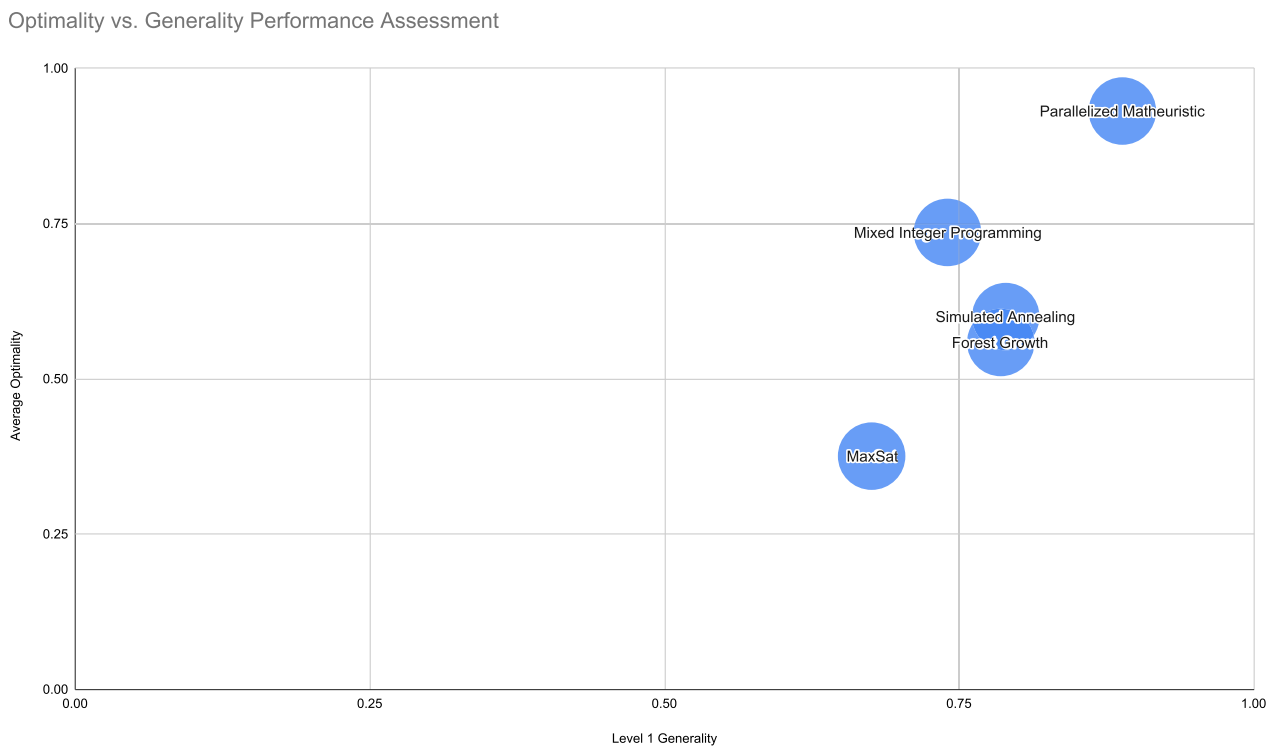


FIGURE 12. **Optimality vs. Generality performance evaluation of literature methods.**

and promoting more general approaches. We encourage future research to utilize the proposed evaluation metrics in their experimental studies in order to facilitate fair comparisons. Furthermore, we propose the incorporation of this framework in future ETP competitions and benchmarks to enhance the rigor and relevance of assessments within the field. Most importantly, as this work aims to align academic research with practical applications of ETPs, applying existing methods to real-world scenarios and assessing their performance using the proposed framework would be highly valuable. Furthermore, it would be interesting to see this framework adopted in solvers designed for real-world problems, such as UniTime.

An interesting future research direction would be the optimality vs. generality performance assessment of a single method across various parameter settings, consisting of a version with high exploitation, a version with high exploration, and a version with a balanced exploration/exploitation ratio. In extreme parameter values, we expect the first and the second versions to achieve higher optimality scores for smaller and larger instances, respectively, with the balanced version behaving more consistently across all instances. Under rare parameter values, we anticipate all three versions maintaining similar average optimality scores. In such a scenario, we predict the balanced version attaining considerably higher level 1 generality. While replicating this precise scenario might be challenging and time-consuming, observing the validity of these predictions would offer valuable insights.

A promising avenue for future research lies in exploring the performance of meta-heuristics with adaptive parameter tuning mechanisms in ETPs utilizing our proposed framework. These methods dynamically adjust their parameters based on the problem instance's characteristics and size, potentially achieving high optimality and level 1 generality scores.

The proposed performance assessment framework is developed for meta-heuristics within the scope of ETPs. This scope limitation constrains its adaptability to minimizing optimization problems using the defined total cost formulation. While the concepts of optimality and generality are problem-independent, the total cost and fitness must be redefined for problems outside this scope. Future research can focus on evaluating the effectiveness of this framework in other contexts with different cost functions. Moreover, the proposed level 1 generality performance metric gives equal attention to all problem instances and penalizes deviations in optimality equally. In problems where certain instances carry greater importance, the generality metric can be redefined in a weighted manner, assigning higher penalties to deviations in more critical instances.

Beyond meta-heuristics in ETPs, future research can explore adopting our proposed framework in other

approaches within the field of OR. The concepts of generality and performance consistency, often overlooked in the literature, are highly essential and insightful in COPs. We believe this research gap deserves much more investigation across different problem domains to advance our understanding and development of robust, generalizable optimization techniques.

Finally, our level 1 generality performance metrics can easily be extended to higher generality orders. Level 2 generality is derived from the SD of level 1 generality values across different datasets, while level 3 generality is calculated from the SD of level 2 generality values across various problems, and so forth. This provides the opportunity to investigate the performance of meta-heuristics over different datasets, or hyper-heuristics over different problems. Furthermore, it provides a better understanding of the optimality vs. generality trade-off and the implications of the NFL theorem.

ACKNOWLEDGMENT

The authors would like to thank Ellenesse Chiew Poh Wen, for her assistance in creating the figures for this work. Their thanks also go to Dr. Dennis Søren Holm from the Data Science for University Management (DSUM) for generously dedicating his time to answer their questions and providing the processed datasets used in their research [70], [71], [72], [73]. They are also grateful to Edon Gashi for his generosity in addressing their queries, organizing personal meetings, and offering clarifications regarding their source code and research [76], [77]. They also extend their thanks to Dr. Tomáš Müller [19], [61] for his assistance in clarifying their questions about the ITC 2019 and for enabling their use of online validator, which facilitated this research. Finally, they extend their appreciation to the reviewers for their detailed comments and constructive suggestions during the peer review process.

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.

[2] E. S. K. Siew, S. N. Sze, S. L. Goh, G. Kendall, N. R. Sabar, and S. Abdullah, "A survey of solution methodologies for exam timetabling problems," *IEEE Access*, vol. 12, pp. 41479–41498, 2024.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] E. Steiner, U. Pferschy, and A. Schaerf, "Curriculum-based university course timetabling considering individual course of studies," *Central Eur. J. Oper. Res.*, vol. 33, no. 1, pp. 277–314, Mar. 2025.

[8] S. Abdipoor, R. Yaakob, S. L. Goh, and S. Abdullah, "Meta-heuristic approaches for the university course timetabling problem," *Intell. Syst. Appl.*, vol. 19, Sep. 2023, Art. no. 200253.

[9] C. H. Wong, S. L. Goh, and J. Likoh, "A genetic algorithm for the real-world university course timetabling problem," in *Proc. IEEE 18th Int. Colloq. Signal Process. Appl. (CSPA)*, May 2022, pp. 46–50.

[10] 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.

[11] K. Xiang, X. Hu, M. Yu, and X. Wang, "Exact and heuristic methods for a university course scheduling problem," *Expert Syst. Appl.*, vol. 248, Aug. 2024, Art. no. 123383.

[12] E. K. Burke, E. K. Burke, G. Kendall, and G. Kendall, *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Cham, Switzerland: Springer, 2014.

[13] R. Lewis and J. Thompson, "Analysing the effects of solution space connectivity with an effective Metaheuristic for the course timetabling problem," *Eur. J. Oper. Res.*, vol. 240, no. 3, pp. 637–648, Feb. 2015.

[14] T. Thepphakorn, P. Pongcharoen, and S. Vitayasak, "Multi-objective hybrid optimizations for designing course schedules based on operating costs and resource utilization," *Ann. Operations Res.*, vol. 2024, pp. 1–68, Nov. 2024.

[15] X. Shao, F. S. Kshitij, and C. S. Kim, "GAILS: An effective multi-object job shop scheduler based on genetic algorithm and iterative local search," *Sci. Rep.*, vol. 14, no. 1, p. 2068, Jan. 2024.

[16] U. R. K. Widayu, A. Mukhlason, and I. Nurkasanah, "Automation and optimization of course timetabling using the iterated local search hyper-heuristic algorithm with the problem domain from the 2019 international timetabling competition," in *Proc. 3rd East Indonesia Conf. Comput. Inf. Technol. (EIConCIT)*, Apr. 2021, pp. 134–138.

[17] 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.

[18] N. Pillay and R. Qu, "Assessing hyper-heuristic performance," *J. Oper. Res. Soc.*, vol. 72, no. 11, pp. 2503–2516, Nov. 2021.

[19] 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-)*, Jan. 2018, pp. 5–31.

[20] N. R. Sabar, S. L. Goh, A. Turky, and G. Kendall, "Population-based iterated local search approach for dynamic vehicle routing problems," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 2933–2943, Oct. 2022.

[21] H. Xiong, S. Shi, D. Ren, and J. Hu, "A survey of job shop scheduling problem: The types and models," *Comput. Oper. Res.*, vol. 142, Jun. 2022, Art. no. 105731.

[22] C. M. Ngoo, S. L. Goh, S. N. Sze, N. R. Sabar, S. Abdullah, and G. Kendall, "A survey of the nurse rostering solution methodologies: The state-of-the-art and emerging trends," *IEEE Access*, vol. 10, pp. 56504–56524, 2022.

[23] J. McCarthy, "Generality in artificial intelligence," *Commun. ACM*, vol. 30, no. 12, pp. 1030–1035, 1987.

[24] A. C. Kumari and K. Srinivas, "Hyper-heuristic approach for multi-objective software module clustering," *J. Syst. Softw.*, vol. 117, pp. 384–401, Jul. 2016.

[25] A. Sosa-Ascencio, G. Ochoa, H. Terashima-Marin, and S. E. Conant-Pablos, "Grammar-based generation of variable-selection heuristics for constraint satisfaction problems," *Genetic Program. Evolvable Mach.*, vol. 17, no. 2, pp. 119–144, Jun. 2016.

[26] R. Raghavjee and N. Pillay, "A genetic algorithm selection perturbative hyper-heuristic for solving the school timetabling problem," *ORiON*, vol. 31, no. 1, p. 39, Apr. 2015.

[27] 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.

[28] N. Pillay, "Evolving hyper-heuristics for the uncapacitated examination timetabling problem," *J. Oper. Res. Soc.*, vol. 63, no. 1, pp. 47–58, Jan. 2012.

[29] K. Z. Zamli, B. Y. Alkazemi, and G. Kendall, "A Tabu search hyper-heuristic strategy for t-way test suite generation," *Appl. Soft Comput.*, vol. 44, pp. 57–74, Jul. 2016.

[30] E. López-Camacho, H. Terashima-Marin, P. Ross, and G. Ochoa, "A unified hyper-heuristic framework for solving bin packing problems," *Expert Syst. Appl.*, vol. 41, no. 15, pp. 6876–6889, Nov. 2014.

[31] H. Terashima-Marín, P. Ross, C. J. Farías-Zárate, E. López-Camacho, and M. Valenzuela-Rendón, "Generalized hyper-heuristics for solving 2D regular and irregular packing problems," *Ann. Oper. Res.*, vol. 179, no. 1, pp. 369–392, Sep. 2010.

[32] 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.

[33] M. Mısır, K. Verbeeck, P. D. Causmaecker, and G. V. Berghe, "A new hyper-heuristic as a general problem solver: An implementation in HyFlex," *J. Scheduling*, vol. 16, no. 3, pp. 291–311, Jun. 2013.

[34] C. Y. Chan, F. Xue, W. H. Ip, and C. F. Cheung, "A hyper-heuristic inspired by pearl hunting," in *Proc. Int. Conf. Learn. Intell. Optim.*, Jan. 2012, pp. 349–353.

[35] T. Cichowicz, M. Drozdowski, M. Frankiewicz, G. Pawlak, F. Rytwiński, and J. Wasilewski, "Five phase and genetic hive hyper-heuristics for the cross-domain search," in *Proc. Int. Conf. Learn. Intell. Optim.*, Paris, France, Jan. 2012, pp. 354–359.

[36] R. P. Badoni, J. Sahoo, S. Srivastava, M. Mann, D. K. Gupta, S. Verma, P. S. Stanimirović, L. A. Kazakovtsev, and D. Karabašević, "An exploration and exploitation-based Metaheuristic approach for university course timetabling problems," *Axioms*, vol. 12, no. 8, p. 720, Jul. 2023.

[37] M. C. Chen, S. N. Sze, S. L. Goh, and S. P. Lau, "Investigation of heuristic orderings with a perturbation for finding feasibility in solving real- world university course timetabling problem," in *Proc. Int. Conf. Digit. Transformation Intell. (ICDI)*, Dec. 2022, pp. 168–173.

[38] 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.

[39] A. A. Gozali, B. Kurniawan, W. Weng, and S. Fujimura, "Solving university course timetabling problem using localized island model genetic algorithm with dual dynamic migration policy," *IEEJ Trans. Electr. Electron. Eng.*, vol. 15, no. 3, pp. 389–400, Mar. 2020.

[40] 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.

[41] C. W. Fong, H. Asmuni, and B. McCollum, "A hybrid swarm-based approach to university timetabling," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 870–884, Dec. 2015.

[42] E. K. Chong and S. H. Żak, *An Introduction to Optimization*, vol. 76. Hoboken, NJ, USA: Wiley, 2013.

[43] M. Mısır, K. Verbeeck, P. De Causmaecker, and G. Vanden Berghe, "An investigation on the generality level of selection hyper-heuristics under different empirical conditions," *Appl. Soft Comput.*, vol. 13, no. 7, pp. 3335–3353, Jul. 2013.

[44] C. Park, Y. Mei, S. Nguyen, G. Chen, and M. Zhang, "Investigating the generality of genetic programming based hyper-heuristic approach to dynamic job shop scheduling with machine breakdown," in *Proc. Australas. Conf. Artif. Life Comput. Intell.*, Geelong, VIC, Australia. Cham, Switzerland: Springer, Dec. 2016, pp. 301–313.

[45] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Cham, Switzerland: Springer, 2015.

[46] D. H. Wolpert and W. G. Macready, "No free lunch theorems for search," Santa Fe Inst., Tech. Rep. SFI-TR-95-02-010, pp. 2756–2760, 1995, vol. 10, no. 12.

[47] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[48] D. H. Wolpert, "What is important about the no free lunch theorems?" in *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems*. Cham, Switzerland: Springer, 2021, pp. 373–388.

[49] S. P. Adam, S.-A. N. Alexandropoulos, P. M. Pardalos, and M. N. Vrahatis, "No free lunch theorem: A review," in *Approximation and Optimization: Algorithms, Complexity and Applications*, vol. 145, I. Demetriou and P. Pardalos, Eds., Cham, Switzerland: Springer, 2019. [Online]. Available: https://doi.org/10.1007/978-3-030-12767-1_5

[50] T. Joyce and J. M. Herrmann, "A review of no free lunch theorems, and their implications for Metaheuristic optimisation," in *Nature-Inspired Algorithms and Applied Optimization*, vol. 744, X. S. Yang, Ed., Cham, Switzerland: Springer, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-67669-2_2

[51] J. He and X. Yao, "Drift analysis and average time complexity of evolutionary algorithms," *Artif. Intell.*, vol. 127, no. 1, pp. 57–85, Mar. 2001.

[52] J. He and X. Yao, "From an individual to a population: An analysis of the first hitting time of population-based evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 6, no. 5, pp. 495–511, Oct. 2002.

[53] T.-L. Yu, K. Sastry, D. E. Goldberg, and M. Pelikan, "Population sizing for entropy-based model building in genetic algorithms," Dept. Illinois Genetic Algorithms Lab., Univ. Illinois, Champaign, IL, USA, Tech. Rep. 2006020, 2006.

[54] D. E. Goldberg, K. Sastry, and T. Latoza, "On the supply of building blocks," in *Proc. 3rd Annu. Conf. Genetic Evol. Comput.*, Jul. 2001, pp. 336–342.

[55] G. Harik, E. Cantú-Paz, D. E. Goldberg, and B. L. Miller, "The Gambler's ruin problem, genetic algorithms, and the sizing of populations," *Evol. Comput.*, vol. 7, no. 3, pp. 231–253, Sep. 1999.

[56] K. Sastry and D. E. Goldberg, "Designing competent mutation operators via probabilistic model building of neighborhoods," in *Proc. Genetic Evol. Comput. Conf.*, Jan. 2004, pp. 114–125.

[57] T. Chen, K. Tang, G. Chen, and X. Yao, "A large population size can be unhelpful in evolutionary algorithms," *Theor. Comput. Sci.*, vol. 436, pp. 54–70, Jun. 2012.

[58] B. Minnaert, D. Martens, M. De Backer, and B. Baesens, "To tune or not to tune: Rule evaluation for Metaheuristic-based sequential covering algorithms," *Data Mining Knowl. Discovery*, vol. 29, no. 1, pp. 237–272, Jan. 2015.

[59] T. Weise, Y. Wu, R. Chiong, K. Tang, and J. Lässig, "Global versus local search: The impact of population sizes on evolutionary algorithm performance," *J. Global Optim.*, vol. 66, no. 3, pp. 511–534, Nov. 2016.

[60] O. Roeva, S. Fidanova, and M. Paprzycki, "Population size influence on the genetic and ant algorithms performance in case of cultivation process modeling," in *Proc. Recent Adv. Comput. Optim., Results Workshop Comput. Optim.*, Dec. 2014, pp. 107–120.

[61] T. Müller, H. Rudová, and Z. Müllerová, "Real-world university course timetabling at the international timetabling competition 2019," *J. Scheduling*, vol. 2024, pp. 1–21, Apr. 2024.

[62] S. Ghafouri, S. Abdipoor, and J. Doyle, "Smart-kube: Energy-aware and fair kubernetes job scheduler using deep reinforcement learning," in *Proc. IEEE 8th Int. Conf. Smart Cloud (SmartCloud)*, Sep. 2023, pp. 154–163.

[63] M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp, "A racing algorithm for configuring Metaheuristics," in *Proc. Gecco*, Jul. 2002, pp. 11–18.

[64] C. F. M. Toledo, L. Oliveira, and P. M. França, "Global optimization using a genetic algorithm with hierarchically structured population," *J. Comput. Appl. Math.*, vol. 261, pp. 341–351, May 2014.

[65] M. Pelikan and M. Pelikan, *Hierarchical Bayesian Optimization Algorithm*. Cham, Switzerland: Springer, 2005.

[66] F. Kang, J. Li, and Q. Xu, "Structural inverse analysis by hybrid simplex artificial bee colony algorithms," *Comput. Struct.*, vol. 87, nos. 13–14, pp. 861–870, Jul. 2009.

[67] N. Liu, J.-S. Pan, C. Sun, and S.-C. Chu, "An efficient surrogate-assisted quasi-affine transformation evolutionary algorithm for expensive optimization problems," *Knowl.-Based Syst.*, vol. 209, Dec. 2020, Art. no. 106418.

[68] C.-S. Chen, H.-W. Hsu, and T.-L. Yu, "Fast algorithm for fair comparison of genetic algorithms," in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2018, pp. 913–920.

[69] R. Tanabe and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 1658–1665.

[70] D. S. Holm, R. Ø. Mikkelsen, M. Sørensen, and T. J. R. Stidsen, "A MIP based approach for international timetabling competition 2019," Abstract Int. Timetabling Competition, 2019. [Online]. Available: https://orbit.dtu.dk/en/publications/a-mip-based-approach-for-international-timetabling-competition-20

[71] D. S. Holm, R. Ø. Mikkelsen, M. Sørensen, and T. J. R. Stidsen, "A graph-based MIP formulation of the international timetabling competition 2019," *J. Scheduling*, vol. 25, no. 4, pp. 405–428, Aug. 2022.

[72] R. Ø. Mikkelsen and D. S. Holm, "A parallelized matheuristic for the international timetabling competition 2019," *J. Scheduling*, vol. 25, no. 4, pp. 429–452, Aug. 2022.

[73] D. S. Holm, "Data reductions for the international timetabling competition 2019 problem," Tech. Univ. Denmark, 2022. [Online]. Available: https://doi.org/10.11581/DTU.00000241 and https://orbit.dtu.dk/en/publications/data-reductions-for-the-international-timetabling-competition-201

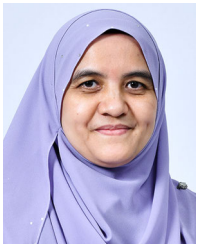
[74] E. Rappos, E. Thiémard, S. Robert, and J. Hêche, "International timetabling competition 2019: A mixed integer programming approach for solving university timetabling problems," in *Proc. Int. Timetabling Competition*, 2019, pp. 1–4.
[75] E. Rappos, E. Thiémard, S. Robert, and J.-F. Hêche, "A mixed-integer programming approach for solving university course timetabling problems," *J. Scheduling*, vol. 25, no. 4, pp. 391–404, Aug. 2022.
[76] E. Gashi, K. Sylejmani, and A. Ymeri, "Simulated annealing with penalization for university course timetabling," in *Proc. 13th Int. Conf. Pract. Theory Automated Timetabling (PATAT)*, Jul. 2022, pp. 361–366.
[77] K. Sylejmani, E. Gashi, and A. Ymeri, "Simulated annealing with penalization for university course timetabling," *J. Scheduling*, vol. 26, no. 5, pp. 1–21, Oct. 2022.
[78] K. Er-rhaimini, "Forest growth optimization for solving timetabling problems," in *Proc. Int. Timetabling Competition*, 2019. [Online]. Available: https://www.itc2019.org/home
[79] A. Lemos, P. T. Monteiro, and I. Lynce, "ITC-2019: A maxsat approach to solve university timetabling problems," Tech. Rep., 2019.
[80] A. Lemos, P. T. Monteiro, and I. Lynce, "Introducing UniCorT: An iterative university course timetabling tool with MaxSAT," *J. Scheduling*, vol. 25, no. 4, pp. 371–390, Aug. 2022.
[81] Y. Censor, "Pareto optimality in multiobjective problems," *Appl. Math. Optim.*, vol. 4, no. 1, pp. 41–59, Mar. 1977.

SINA ABDIPOOR received the bachelor's degree in software engineering and the master's degree in AI. He is a Ph.D. Researcher in artificial intelligence (AI) with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. Currently, his research efforts are concentrated on the application of hybrid approaches for tackling real-world scheduling problems. Beyond academia, he has extensive experience working as an AI Engineer and consultant in various domains. He frequently discusses the applications of OR theory in real-world problems to save time and improve efficiency. His primary research interests include operations research (OR), where he specializes in utilizing meta-heuristics to address various optimization problems. He is an active member of Canadian Operational Research Society and is passionate about advancing AI and its practical applications. In addition to his active role as a Researcher, he serves as a Reviewer for *Journal of Expert Systems with Applications* (Elsevier), *Intelligent Systems with Applications* (Elsevier), and IEEE Access. For more details please visit https://sites.google.com/view/sina-abdipoor

RAZALI YAAKOB received the bachelor's and master's degrees in computer science from the Universiti Putra Malaysia, in 1996 and 1999, respectively, and the Ph.D. degree from the University of Nottingham, U.K., in 2008. Currently, he is an Associate Professor with the Faculty of Computer Science and IT, Universiti Putra Malaysia. His research interests include artificial neural networks, pattern recognition, evolutionary computation, and deep learning. He is a member of the Intelligent Computing Group at the faculty.

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

SALWANI ABDULLAH received the B.Sc. degree in computer science from the University of Technology Malaysia, the master's degree specializing in computer science from the National University of Malaysia, and the Ph.D. degree in computer science from the University of Nottingham, U.K. Currently, she is a Professor of computational optimization with the Faculty of Information Science and Technology, National University of Malaysia. Her research interests include artificial intelligence and operation research, particularly computational optimization algorithms (heuristic and meta-heuristic, evolutionary algorithms, and local search) that involve different real-world applications for single and multi-objective continuous and combinatorial optimization problems, such as timetabling, scheduling, routing, nurse rostering, dynamic optimization, data mining problems (feature selection, clustering, classification, time-series prediction), intrusion detection, and search-based software testing.



HAZLINA HAMDAN received the B.Sc. degree (Hons.) in computer science from the Universiti Kebangsaan Malaysia, the Master of Computer Science degree in artificial intelligence from the Universiti Malaya, and the Ph.D. degree from the University of Nottingham, U.K. She is currently a Senior Lecturer with the Department of Computer Science, Universiti Putra Malaysia. Her research interests include intelligent computing and application, such as medical prognostic, pattern recognition, prediction systems, and optimization.

KHAIRUL AZHAR KASMIRAN received the Ph.D. degree from The University of Sydney, Australia, in 2012. He is a Senior Lecturer with the Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia. His research interests include deep learning, reinforcement learning, performance engineering, formal verification, and software development.