Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A survey of the state-of-the-art of optimisation methodologies in school timetabling problems

Joo Siang Tan [a], Say Leng Goh [a,*], Graham Kendall [b,c], Nasser R. Sabar [d]

[a] Optimisation Research Group, Faculty of Computing & Informatics, Universiti Malaysia Sabah Kampus Antarabangsa Labuan, Jln Sugai pagar, 87000 Labuan, Malaysia
[b] School of Computer Science, The University of Nottingham Malaysia Campus, Jln Broga, 43500 Semenyih, Selangor, Malaysia
[c] School of Computer Science, University of Nottingham, University Park, Nottingham NG7 2RD, United Kingdom
[d] Department of Computer Science and Information Technology, La Trobe University, Melbourne, Australia

ARTICLE INFO

Keywords:
School timetabling
Educational timetabling
Meta-heuristic algorithms

ABSTRACT

Educational timetabling is an ongoing challenging administrative task that is required in most academic institutions. This is mainly due to a large number of constraints and requirements that have to be satisfied. Educational timetabling problems have been classified as NP-hard problems and can be divided into three types: exam timetabling, course timetabling and high school timetabling. The domain of high school timetabling is not well developed when compared to other fields of educational timetabling such as university exam timetabling and course timetabling. As the evolution of the educational systems are continuous, new challenges often arise, requiring new models and solution methodologies. Over the years, a number of methodologies have been developed to address high school timetabling problems. However, there are no comparative studies or rigorous analysis of these methodologies. This survey paper aims to provide a scientific review of high school timetabling. The paper presents a categorisation of the methodologies conducted in recent years based on chronology, category and application (dataset). We first present comparative studies on the success of proposed methodologies. The components and mechanisms of different methodologies are analysed and compared. We also discuss their performance, advantages, disadvantages and potential for improvement. Methodology wise, a shift of popularity from meta-heuristic to mathematical optimisation is observed in recent years. Another observation is that more researchers are opting for XHSTT formatted datasets as a testbed for their algorithms. Finally, we outline the industrial perspective, trends and future direction in high school timetabling optimisation problems.

1. Introduction

A timetabling problem is composed of four parameters: a finite set of times (T), a finite set of resources (R), a finite set of meetings (M) and a finite set of constraints (C). The objective is to assign times and resources to the meetings by minimising the constraint violations (Qu et al., 2009). There are three main types of educational timetabling problems: course, examination and high school. The course timetabling problem involves assigning a set of courses to finite time-slots and featured rooms (Akkan & Gulcu, 2018; Babaei et al., 2015; Deris et al., 2000; Goh et al., 2019; Lewis, 2008; Zheng et al., 2015). The examination timetabling problem involves assigning a set of exams into a set of timeslots and capacitated rooms (Leite et al., 2019; Sabar et al., 2014b, 2012; Woumans et al., 2016). The high school timetabling problem covers assigning resources such as times, teachers, students and rooms to a collection of events (Fonseca et al., 2016b). Timetabling problems are well-known challenging combinatorial optimisation problem and have been classified as NP-hard problems (Pillay, 2014).

This paper focuses on high school timetabling problems. Due to different cultural settings and educational systems, high school timetabling problems differ for each country in terms of requirements and timetabling structure. As the evolution of the education system is continuous, new problems with new requirements constantly appear requiring new solution methodologies (Post, Ahmadi, Daskalaki et al., 2012). Several researchers including (Kristiansen & Stidsen, 2013; Pillay, 2014; Schaerf, 1999b) have discussed approaches used in solving high school timetabling problem. However, there are no comparative studies on the success of methodologies, their performance, advantages, disadvantages and potential improvements. To this end, this paper

* Correspondence to: Optimization Research Group, Faculty of Computing and Informatics, Universiti Malaysia Sabah Kampus Antarabangsa , Labuan, 87000, Malaysia.

E-mail addresses: jsiang@live.com (J.S. Tan), slgoh@ums.edu.my (S.L. Goh), graham.kendall@nottingham.edu.my (G. Kendall), n.sabar@latrobe.edu.au (N.R. Sabar).

https://doi.org/10.1016/j.eswa.2020.113943
Received 5 May 2020; Received in revised form 7 August 2020; Accepted 29 August 2020
Available online 1 September 2020

presents an updated comprehensive review and categorisation of the latest research works for high school timetabling problem. Our aims are:

- To provide the definition, terminologies and constraints involved in the high school timetabling problem to promote a greater understanding in this domain.
- To present a comprehensive technical review and categorisation of the latest methodologies applied to this domain.
- To outline the benchmark datasets for high school timetabling problems and the respective state of the art methods.
- To discuss the industrial perspective, trends and future direction in high school timetabling.

The contributions of this paper are as follows:

1. This paper describes the most recent works in high school timetabling. The components and mechanisms of different methodologies are analysed and compared. In addition, we present comparative studies on the success of those methodologies. To the best of our knowledge, no previous survey of high school timetabling problem covers these fields.
2. We arrange the solution techniques and approaches in chronological order irrespective of datasets utilised by the researchers, in order to observe the trends and popularity of methodologies in this domain.
3. We also divide the solution techniques and approaches into four major categories namely meta-heuristics, hyper-heuristics, mathematical optimisation and matheuristics. We discuss their performance, advantages, disadvantages and potential for improvements.
4. We present the various standard data formats proposed by researchers to accommodate a fair comparison of algorithms. We also discuss the features of the existing benchmark datasets as well as their use in recent years as a standard test-bed. Finally, we group the solution techniques and approaches based on datasets and present the respective state of the art methodologies.

The following section provides a general view of high school timetabling problem. Section 2 discusses problem definition, problem constraints and the mathematical formulation. An overview of the latest methodologies applied in high school timetabling are presented in Section 3. Section 4 presents the categorisation of the survey solution methods. Section 5 shows the data format available for high school timetabling. Section 6 summarises the commercial software and freeware that is available. Lastly, future directions and conclusions are given in Sections 7 and 8.

2. High school timetabling problem

2.1. Problem definition

High school timetabling problems can be defined in terms of the availability of teachers and rooms, a number of lessons to be taught by teachers to specific classes or students and a set of constraints. The construction of a timetable involves assigning resources, such as times, teachers, students and rooms to a collection of lessons while minimising the constraint violations (Fonseca et al., 2016b; Sørensen & Dahms, 2014; Tassopoulos et al., 2020). For example, teachers must teach specific lessons, in particular rooms, to specific classes or students at specific times. Constraints are a set of conditions that should be satisfied in a solution, and they are categorised into two types, hard and soft. Hard constraints must be satisfied for a solution to be valid, whereas soft constraints can be violated but assigned a given penalty (Schaerf, 1999a). A feasible timetable must not violate any hard constraints, and the quality of a timetable is measured based on the number of soft constraints satisfied, where each soft constraint violation may be weighted differently.

2.2. Terminology

This section describes the terms used in this paper.

- *Class*. Refers to a group of students that attend a particular subject simultaneously.
- *Lesson*. Refers to a period of learning or teaching (e.g. a particular subject being taught to a class by a teacher)
- *Time group*. Comprises a set of time slots, and it usually refers to a day or a week.
- *Time slot*. A specific duration in which a lesson can be scheduled in a timetable.
- *Event*. Actual content (e.g. English subject) that could split into several sub-events or lessons.
- *Event group*. Refers to a group of events with a particular similarity (e.g. a set of lessons taught by the same teacher in different time slots.)
- *Resource*. Refers to a teacher, student, class or room.
- *Resource group*. Refers to a group of teachers, students, classes or rooms.
- *Role*. An identity used to identify a resource in an event (e.g. an event contains two roles, and each role is used to determine the suitable resource during resource assignment).

2.3. Problem constraints

Post, Ahmadi, Daskalaki et al. (2012) and Post et al. (2016) reviewed several high school timetabling problems in different countries. They summarised a set of common constraints that are widely used and categorised the constraints into three groups: basic scheduling problem, constraints for the event, and constraints for resources. Pillay (2014) further split the groups into seven groups: problem requirement constraints, no clash constraints, resources utilisation constraints, workload constraints, period distribution constraints, preference constraints and lesson constraints. Here, we present some specific constraints so that readers can have a better understanding of the problem. The constraints which can be hard or soft depending on the dataset and/or user requirements;

- C1: Assign a time slot to the selected event.
- C2: Assign a resource to the selected event with a specific role.
- C3: Schedule the selected resources or resource group without clashes (no resource attends two events for the same time).
- C4: Split the selected event or event group into sub-events, within a maximum and a minimum number of amount and duration.
- C5: Limit the workload of the selected resources or resource groups.
- C6: Two subevents must be consecutive when scheduled for the same day, in case it is required by the event.
- C7: Each class, for a given period, is assigned to one teacher at most, except for the case of co-teaching.
- C8: A selected event cannot be assigned to any of its forbidden time slots.
- C9: Assign a preferred time slot to a selected event.
- C10: Assign a preferred resource to a selected event with a specific role.
- C11: Schedule the selected event groups at the same time slot.
- C12: Schedule the selected event groups to a selected time group between a minimum and a maximum number of time slots.
- C13: Schedule the selected event groups with a specific role to the same resource.
- C14: Split the selected event or event group into sub-events with a specific duration, within a maximum and a minimum number of sub-events.
- C15: Assign time slot in chronological order to selected events.
- C16: Avoid the selected resources are busy in the selected time slot.

- C17: Each selected resources or resource groups should have a limited number of idle time slots in selected time groups.
- C18: Each selected resources or resource groups should have a limited number of occupied time slots in the selected time groups.
- C19: Each selected resources or resource groups should have a limited number of time groups that contain at least an assigned time slot.
- C20: Each selected event that fulfilled with its specific requirement should have at least a certain number of double lessons (two constructive periods of sub-events) in a week.
- C21: Minimise the number of occurrences of two events of the same class being assigned two consecutive days.
- C22: A class must not be scheduled to attend more than a certain number of lessons with the same teacher per day.

3. Solution techniques and approaches

This section reviews the state-of-the-art of algorithms applied to the high school timetabling problem. Existing algorithms can be classified into one or more of the following types:

- **Mathematical optimisation algorithms**. Mathematical algorithms such as Integer Programming and Constraint Programming assume the objective function and the constraints are linear and restrict some or all of the problem variables into integer values.
- **Meta-heuristic algorithms**. Meta-heuristic algorithms are very general purpose problem-solvers designed to find an acceptable solution in a reasonable amount of computational time. Example of these algorithms are:
    - Population-based algorithms: maintain and work on set of candidate solutions in which each solution corresponding to a point in the problem search space.
    - single solution-based algorithms: maintain a single candidate solution and use move operator to explore the area around the current solution.
- **Graph colouring algorithms**. Graph colouring algorithms use graph theory to represent the problem variables.
- **Matheuristics approaches**. Matheuristics approaches combine the strengths of heuristic method with the mathematical optimisation algorithms.
- **Hyper-heuristic approaches**. Hyper-heuristic approaches use a set of heuristic and a selection method to automate the selection of which heuristic should be applied.
- **Hybrid approaches**. Hybrid approaches combine the strengths of several (two or more) meta-heuristic algorithms in a unified framework.

3.1. Integer programming

Sørensen and Dahms (2014) proposed a two-stage decomposition of Integer Programming (IP) model to solve the Lectio school timetabling problem. The model is based on their previous IP model (Sorensen & Stidsen, 2013). They used decomposition to split the model into two stages and removed unnecessary variables to decrease the execution time. In the first stage, events were assigned into timeslots. In the second stage, events were assigned to rooms. To overcome the irreversible decision made in stage one, the stage two model is partly incorporated in the stage one model by exploiting minimum weight and maximum matching problem in the bipartite graph. The computational results show a significant improvement in their model compared to the previous IP model.

Kristiansen et al. (2015) proposed a Mixed-Integer Linear Programming (MILP) formulation for XHSTT formatted dataset. Their approach consists of two steps. In the first step, the MIP model was built with hard constraints only. It is used to minimise hard constraint violations until the time limit was reached or a feasible solution was found. If a feasible solution was found, the second step was invoked. The solution was further improved in the remaining time by adding all the soft constraints into the model. The approach was able to find optimal solutions for two instances and prove optimality of four known solutions.

Al-Yakoob and Sherali (2015) proposed a two-stage approach (TSA) of Mixed Integer Programming model (MIP), and an alternative method of Column Generation Heuristic (CGH) for MIP to six high school timetabling problems from Kuwait. In the first stage, denoted as model 1, weekly timeslots were determined for the classes. Teachers were assigned to classes in the second stage, denoted as model 2. For the alternative method, they presented a comprehensive MIP (denoted as model 3) to select valid combinations of weekly schedules for each employed teacher. Due to the exponential number of variables in model 3, a CGH was designed to solve its linear programming relaxation, where a sequential variable-fixing heuristic is created to produce a good quality feasible solution. Both approaches produced solutions in relatively short CPU times, but the CGH outperformed the TSA.

Fonseca et al. (2017) further improved Kristiansen's work (Kristiansen et al., 2015) by employing new cuts and reformulations for the existing IP model to solve the ITC2011 dataset. From the original formulation, Fonseca improved some valid inequalities and an extended flow-based formulation. As a result, the fractional points were reduced in the new formulation. Pre-processing routines were proposed to remove unnecessary constraints and variables. A multi-commodity flow reformulation was used to drop several non-essential constraints to reduce the size of the constraints in an alternative formulation. The proposed cuts improved the linear relaxation of the formulation, leading to an average gap reduction of 32%. It is the current state-of-the-art for the problem instances in ITC2011.

Dorneles et al. (2017) proposed an IP based on the multi-commodity flow model for Brazilian high school timetabling problems. Each commodity represents a teacher, and the arcs represent transitions of time periods in a particular network graph. Dantzig–Wolfe decomposition principles are applied to obtain an alternative formulation. Then a column generation approach is used to solve linear relaxation of the alternative formulation. The computational experiment showed the performance of their approach was faster than previous approaches (Dorneles et al., 2014) reported in the scientific literature. Moreover, five lower bounds out of 12 instances were improved.

Tassopoulos et al. (2020) proposed a Mixed-Integer Programming (MIP) model to solve high school timetabling problems in Greece. The model was implemented using Gurobi and CPLEX. Two methodologies were employed in the formulation. The first utilised a "monolithic" model that included all hard and soft constraints. The second decomposed the problem into six sub-problems. The computational results showed the second method was effective in producing optimal solutions while the first method did not produce satisfactory results.

3.2. Adaptive Large Neighbourhood Search (ALNS)

Sørensen et al. (2012) applied ALNS to the ITC2011 dataset, and the algorithm was among the round two finalists in the competition. Three strategies were implemented in ALNS: Remove Strategy (removal and insertion methods specifically for sub-events), Adaptive Strategy (a metric to influence the selection of methods in the remove strategy) and Accept Strategy (acceptance criteria similar to SA). The results were competitive in most cases compared to the best-known solutions.

Sørensen and Stidsen (2012) proposed ALNS for six Danish high school timetabling problems in the Lectio dataset. In their implementation, two neighbourhood operators were used; namely insertion and removal. The insertion operator consists of three functions: InsertGreedy, InsertRegretN and RoomInsert. It was used to insert event chains based on their objective measurement. The removal operator consists of five functions: RemoveRandom, RemoveRelated, RemoveTime, RemoveClass and RoomRemove. It was used to remove or unassign a resource or event based on the objective. The un-assignment of events allowed events to be assigned a time slot and a room.

3.3. *Variable Neighbourhood Search (VNS)*

Saviniec et al. (2013) proposed VNS to solve Brazilian high school timetabling problems by employing two neighbourhood structures: torque (TQ) and matching (MT) operators. The TQ operator was inspired by the chain of moves idea (a graph was built to identify components that caused conflicting meetings) whereas the MT operator is based on the resolution of assignment problems (rescheduling a set of teachers in a class). The statistical distribution was very close to the optimal solution, which was achieved for all instances by using their method.

Fonseca and Santos (2014) proposed several VNS algorithms to solve ITC2011 high school timetabling problems. The variants of VNS tested are basic VNS, Reduced VNS (RVNS), Skewed VNS (SVNS) and Sequential Variable Neighbourhood Descent (SVND). The basic VNS explored the search space by making systematic changes in neighbourhood structures. A key component of VNS algorithms is the descent phase. In RVNS, they removed the descent phase to improve its performance. In SVNS, a neighbourhood structure is considered in each iteration and they had a relaxed rule to accept the candidate solution. In SVND, the search space is explored by using only one neighbourhood structure per each iteration. The SVNS approach outperformed the winner of ITC2011. Feasible solutions were found for 12 out of 18 instances. In conclusion, SVNS performed the best, while RVNS was the worst performer among all the VNS variants.

Teixeira et al. (2019) proposed two algorithms based on VNS in solving ITC2011 instances. The first algorithm was named Skewed General Variable Neighbourhood Search (SG-VNS), where Variable Neighbourhood Descent (VND) was used as a local search. It used systematic changes of neighbourhood functions to explore the solution space. Intermediate solutions that were worse than its current solution were accepted based on a parameter. The second algorithm was named Adaptive Variable Neighbourhood Search (AVNS). It used adaptive probability selection in selecting neighbourhood functions where the neighbourhood functions that generated better solutions had higher probabilities to be chosen. The computational results showed SGVNS was better than AVNS, and both algorithms produced better results compared to the winning algorithm of ITC2011.

3.4. *Particle Swarm optimisation (PSO)*

Tassopoulos and Beligiannis (2012b) proposed a PSO algorithm in constructing feasible and efficient timetables for high schools in Greece. In the beginning, 150 active particles with different fitness value were initialised. In each generation, each particle was evolved with three procedures (swap with probability, random change with personal best, and random change with global best) to produce a new solution. If the fitness value of a particle exceeded a tolerance value, the particle was inactivated in the evolution procedure until the number of active particles was equal to 30. The purpose was to explore a wider search space and decrease the execution time. After 10,000 iterations, the best solution was further processed by a local search procedure to minimise the teachers' idle time periods. This method outperformed constraint programming and other evolutionary algorithms.

3.5. *Cyclic transfer algorithm*

Post, Ahmadi and Geertsema (2012) proposed a neighbourhood structure based on sequential/cyclic moves and a cyclic transfer algorithm to solve Dutch and English high school timetabling problems. The proposed method improved the solution by enabling the execution of complex moves (a sequence of insertion and ejections). A recursive method was applied to determine an improved cycle by using an improvement graph. The experimental results showed an 8%–28% cost reduction from the initial solution by the proposed method.

3.6. *Constraint programming*

Demirović and Stuckey (2018) proposed a constraint programming (CP) method to the ITC2011 dataset. They formulated the CP model by using a scheduling-based point of view and included a solution-based phase saving technique to direct the first search in close proximity to the best solution found. The initial solution was generated by Kingston High School Timetabling Engine (KHE). They implemented event swap and event block swap techniques to improve the solution. The method was better than the standard CP approach. It outperformed the IP and maxSAT complete methods and provided competitive results compared to VNS (Fonseca & Santos, 2014).

3.7. *Graph colouring*

A two-phase graph edge colouring approach was proposed by Badoni and Gupta (2014) to solve five randomly generated school timetabling problems. A bipartite multi-graph was used to construct daily requirement matrices in the first phase, while a bipartite graph was used to assign lectures over unconstrained time slots in the second phase. The highest degree first heuristic was used for the ordering of the coloured edges, and the computational results demonstrated that it was effective.

3.8. *Parallel local search*

Saviniec et al. (2018) proposed a parallel local search in solving Brazilian high school timetabling problems. The algorithm employed trajectory-based parallel multi-start (ran with several asynchronous threads of trajectory-based methods simultaneously) and manager/workers strategies with shared memories. Two strategies were utilised, namely: (1) central memory-based. (2) diversification and intensification memory-based. The former consists of a group of metaheuristic agents to produce a good solution and kept it in the central memory, while the latter separates the central memory into diversification (maintaining a set of non-elite solutions) and intensification memory (maintaining a set of elite solutions). The method outperformed the state-of-the-art algorithms for variants of the problems.

3.9. *Tabu search*

Minh et al. (2010) presented a Tabu Search (TS) algorithm to solve three high school timetabling problems in Vietnam. Three types of move were utilised namely single moves, swap moves and block-changing moves. An initial solution was created by a greedy algorithm and then improved by the TS. The solutions were better than the ones generated manually.

3.10. *Simulated annealing*

Zhang et al. (2010) proposed a Simulated Annealing (SA) approach with a new neighbourhood structure (a sequence of swap between pairs of timeslots) for the high school timetabling problems. They tested their approach utilising the Greece and OR-library datasets. Their approach consisted of two phases. The first phase was used to generate a feasible solution and the second phase was used to minimise the cost of the solution while maintaining its feasibility. The computational results showed their approach was better than an evolutionary algorithm, constraint programming and column generation.

Odeniyi et al. (2020) proposed an Enhanced Simulated Annealing (ESA) algorithm to address high school timetabling problems in Nigeria. In the preprocessing, mixed integer linear programming technique was utilised to formulate the high school timetabling problem. Three enhancements of ESA were: the modification to the temperature reduction parameter of SA to increase the speed of convergence, integration with a genetic algorithm to improve the quality of solution and reordering the sequence of evolutionary operators (mutation, selection and crossover instead of selection, crossover and mutation). The ESA algorithm managed to obtain optimal solutions in short simulation times compared to the SA approach.

3.11. *Evolutionary algorithms*

Evolutionary algorithms (Andrade et al., 2019; Dutta et al., 2020; Yuan et al., 2020) are population-based methods. They operate on a set of solutions and evolve new ones using selection, crossover and mutation procedures. Due to the problem constraints, few evolutionary algorithms were proposed in the literature. Domrös and Homberger (2012) proposed an evolutionary algorithm for ITC2011, and they were among the finalist. Two techniques are implemented in the solver: an indirect representation of timetable (each encoded solution consisted of a permutation of sub-events) and an evolutionary search controlled by an evolution strategy (a randomised method). In preprocessing, a permutation of eligible sub-events was invoked by taking several constraints into account. In the mutation process, two indices from the permutation were randomly chosen and then swapped. After that, times and required resources were assigned to the eligible sub-events during the decoding procedure.

Raghavjee and Pillay (2013) proposed a two-phased genetic algorithm (GA) to solve the school timetabling problem in Greece and from the OR-library dataset. The first phase produced a feasible solution, while the second phase improved the quality of the solution. Both phases begin by creating an initial population of individuals, and each individual was iteratively improved by mutation (different mutation operators are applied in different phases). The performance was comparable to previous methods (e.g. Simulated Annealing, Neural Network, and Evolutionary Algorithm etc.) applied to the same problems.

Sutar and Bichkar (2016) proposed a Simple Genetic Algorithm (SGA) with knowledge augmented operators and probabilistic repair in crossover to solve the OR-library dataset. A new population was created in each generation by cloning the new offsprings, and the best individual was carried over from the previous population. Domain knowledge was used in the GA operators to guide the search during the construction process. The results showed SGA managed to find all feasible solutions. However, GA with similar operators and crossover method can perform faster.

3.12. *Matheuristics*

Dorneles et al. (2014) proposed a mixed-integer linear programming (MILP) and fix-and-optimise heuristic combined with variable neighbourhood descent (VND) to solve high school timetabling problems in Brazil. The formulations of MIP model were based on the problem constraints. At first, an initial, feasible solution was generated by the MIP model disregarding the objective function, then the solution was further improved by a fix-and-optimise heuristic combined with VND. Three decompositions (class, teacher and day) were proposed in the fix-and-optimise heuristic to allow a certain number of context (classes, teachers, or days) to be optimised. The neighbourhoods created by the fix-and-optimise heuristic were then explored by VNS. The method found best best-known solution for seven out of 12 instances.

Fonseca et al. (2016b) proposed a combination of Variable Neighbourhood Search with mathematical programming-based neighbourhoods to solve ITC2011 high school timetabling problems. The VNS explored the search space by making thematic changes in the neighbourhood structure to find feasible solutions until stagnation. The IP worked as the master controller to control low-level local searches in improving the feasible solutions. In the beginning, an initial solution was produced by KHE, ; then the VNS performed a series of changes to the neighbourhood structures on the initial solution. The IP solver was invoked subsequently to further improve the solution. Their approach outperformed a standalone VNS and was able to improve 15 out of 17 best best-known solutions.

Saviniec et al. (2020) proposed two pattern-based formulations and a cooperative parallel meta-heuristic in solving high school timetabling problems in Brazil. The two pattern-based formulations relied on teachers/classes meeting patterns. The first formulation used the original variables associated with layouts and black-box solver to tackle instances of a practical size. The second formulation used a column generation framework. The parallel meta-heuristic was built on the diversification–intensification memory based framework which is similar to the work (Saviniec et al., 2018) proposed before. New agents were introduced to the parallel framework to exploit the developed column generation method in the second formulation. Their implementation was found to be competitive with the existing formulations (Dorneles et al., 2017).

3.13. *Hyper-heuristics*

Raghavjee and Pillay (2015) proposed a genetic algorithm (GA) selection perturbative hyper-heuristic for school timetabling in Greece and from the OR-library dataset. Two phases were applied in their solver. The first phase focused on hard constraints, while the second phase focused on soft constraints. The second phase was only executed after a feasible solution was found in the first phase. Parents were chosen using a tournament selection. Mutation and crossover operators were applied to produce new offsprings. The method managed to provide feasible solutions for all the instances and performed better than a standard GA.

Ahmed et al. (2015) presented a set of selection hyper-heuristics combining five different selection methods and three move acceptance methods to address the ITC2011 instances. The five different selection methods were: Simple Random (SR), Random Descent (RD), Random Permutation (RP), Random Permutation Descent (RPD) and Choice Function (CF). The three move acceptance methods were: Improve or Equal moves (IE), Great Deluge (GD) and Simulated Annealing (SA). The selection hyper-heuristics were used with a set of nine low level heuristics (seven mutational and two hill climbing heuristics) during the search process. The experimental results revealed their selection method RP embedded with GD move acceptance method outperformed other selection hyper-heuristics. The approach was better than the previous methods e.g. evolutionary algorithm (Domrös & Homberger, 2012) and adaptive large neighbourhood search algorithm (Sørensen et al., 2012) but was inferior to the hybridised Simulated Annealing (SA) and stagnation-free Late Acceptance Hill Climbing (Fonseca et al., 2016a). They concluded that the choice of selection hyper-heuristic and the adaptability of move acceptance components influenced its overall performance.

A stochastic local search algorithm based on selection hyper-heuristic framework was proposed by Kheiri et al. (2016) in ITC2011. The initial solution was constructed by a general solver using KHE library. The solution was then improved by multi-stage selection hyper-heuristic (diversification and intensification stages). The diversification stage employed a selection hyper-heuristic combining simple random heuristic selection with adaptive move acceptance. It used eight low-level mutational heuristics and allowed worse moves to be accepted. The intensification stage used two hill climbing (HC) operators. The first operator was based on ejection chains while the second was a type of first improvement HC operator. The method won second place during the ITC2011 competition.

Table 1
The most recent solution methods for high school timetabling problems (irrespective of datasets).

Year	Method	Author
2010	SimulatedAnnealing	Zhang et al. (2010)
	TabuSearch	Minh et al. (2010)
2012	Adaptive large neighbourhood search	Sørensen et al. (2012)
	Adaptive large neighbourhood search	Sørensen and Stidsen (2012)
	Cyclic transfer algorithm	Post, Ahmadi and Geertsema (2012)
	Evolutionary algorithm	Domrös and Homberger (2012)
	Simulated annealing + Variable neighbourhood search	Brito et al. (2012)
	Particle swarm optimisation	Tassopoulos and Beligiannis (2012b)
	Particle swarm optimisation + Local search	Tassopoulos and Beligiannis (2012a)
2013	Variable neighbourhood search	Saviniec et al. (2013)
	Two-phased genetic algorithm	Raghavjee and Pillay (2013)
2014	Graph colouring	Badoni and Gupta (2014)
	Mixed integer linear programming + Fix-optimise heuristic + Variable neighbourhood descent	Dorneles et al. (2014)
	Two-Stage decomposition integer programming	Sørensen and Dahms (2014)
	Variable neighbourhood search	Fonseca and Santos (2014)
2015	Genetic algorithm selection perturbative hyper-heuristic	Raghavjee and Pillay (2015)
	Mixed integer programming	Kristiansen et al. (2015)
	Particle swarm optimisation + Local search	Katsaragakis et al. (2015)
	Artificial fish swarm + Local search	Katsaragakis et al. (2015)
	Two-stage approach of mixed integer programming model	Al-Yakoob and Sherali (2015)
	Column generation heuristic	Al-Yakoob and Sherali (2015)
	Selection hyper-heuristics	Ahmed et al. (2015)
2016	Simulated annealing + Iterative local search	da Fonseca et al. (2016)
	Simulated annealing + Stagnation-free late acceptance hill climbing	Fonseca et al. (2016a)
	Stochastic local search + Hyper-heuristic framework	Kheiri et al. (2016)
	Simple genetic algorithm	Sutar and Bichkar (2016)
	Variable neighbourhood search + matheuristic	Fonseca et al. (2016b)
2017	Integer programming	Fonseca et al. (2017)
	Iterative local search + Variable neighbourhood search	Saviniec and Constantino (2017)
	Local search + maxSAT-based large neighbourhood search	Demirović and Musliu (2017)
	Multicommodity flow model	Dorneles et al. (2017)
	Sequence-based selection hyper-heuristic	Kheiri and Keedwell (2017)
	Cat Swarm optimisation + Local search	Skoullis et al. (2017)
2018	Constraint programming	Demirović and Stuckey (2018)
	Parallel local search	Saviniec et al. (2018)
2019	Variable neighbourhood search	Teixeira et al. (2019)
2020	Mixed integer programming	Tassopoulos et al. (2020)
	Two-pattern-based formulations + Cooperative parallel meta-heuristic	Saviniec et al. (2020)
	Enhanced simulated annealing	Odeniyi et al. (2020)

Kheiri and Keedwell (2017) proposed a sequence-based selection hyper-heuristic to solve high school timetabling problems in ITC2011. Two components (selection and move acceptance components) were employed in the hyper-heuristic. The first component consists of 15 low-level heuristics. A sequence-based selection was used to select a low-level heuristic to perform the search. The second component is an acceptance criterion which consists of five different meta-heuristic acceptance levels: HC, SA, great deluge (GD), record-to-record travel (RR) and late acceptance (LA). The KHE was used to generate an initial solution and the hyper-heuristic was used to further improve the solution. The empirical results indicated the method was effective in solving the problem compared to the winning algorithm of ITC2011.

3.14. Hybrid approaches

Simulated Annealing (SA) with Iterative Local Search (ILS) was proposed by the GOAL team (da Fonseca et al., 2016) for the ITC2011 dataset. An initial solution was generated by KHE and was improved by SA with reheating. Then ILS was applied to further improve the solution. There were six neighbourhood structures used in SA: Event Swap, Resource Swap, Event Move, Resource Move, Event Block Swap, and Kempe Chain (KC). Reassign Resource Times and KC were used in the perturbation phase of ILS. The method was the winner of ITC2011, and it was successful because of its diversity of local search moves.

Brito et al. (2012) proposed a hybridised Simulated Annealing (SA) and Variable Neighbourhood Search (VNS) to solve high school timetabling problems in ITC2011. In preprocessing, six neighbourhood structures were proposed: Event Swap, Event Move, Event Block Swap, Kempe Move, Resource Swap and Resource Move. They built an initial solution utilising KHE and refined it using SA and VNS in sequence. In SA, a probabilistic selection was employed to select a neighbourhood structure. In VNS, a shake function was used to generate a random neighbouring solution. The approach led them to the finals of the competition by obtaining eight feasible solutions out of 19 instances.

Tassopoulos and Beligiannis improved a PSO by hybridising it with a local search to solve the Greek high school timetabling problems (Tassopoulos & Beligiannis, 2012a). Fifty particles were initialised at a random position in a two-dimensional matrix, and each particle went through 3 procedures (swap function, insertion from personal best, and insertion from global best) to produce a feasible solution until the termination criterion was met. After PSO terminated, the solution was further improved by a local search procedure to minimise soft constraint violations by using five adjustment weights to influence the solution acceptance criteria. Their implementation achieved better results compared to the Evolutionary Algorithm and Simulated Annealing.

Katsaragakis et al. (2015) investigated the performance of hybrid Particle Swarm optimisation (PSO) and hybrid Artificial Fish Swarm (AFS) in addressing high school timetabling problems in Greece. Their PSO consisted of 15 particles where each particle comprised a two-dimensional array. The PSO had several features that distinguished it from the work of Tassopoulos and Beligiannis (2012a). Their AFS consisted of 24 fish, each of which was a two-dimensional array. The AFS has seven basic procedures: Prey, InnerPrey, SwarmNChance,

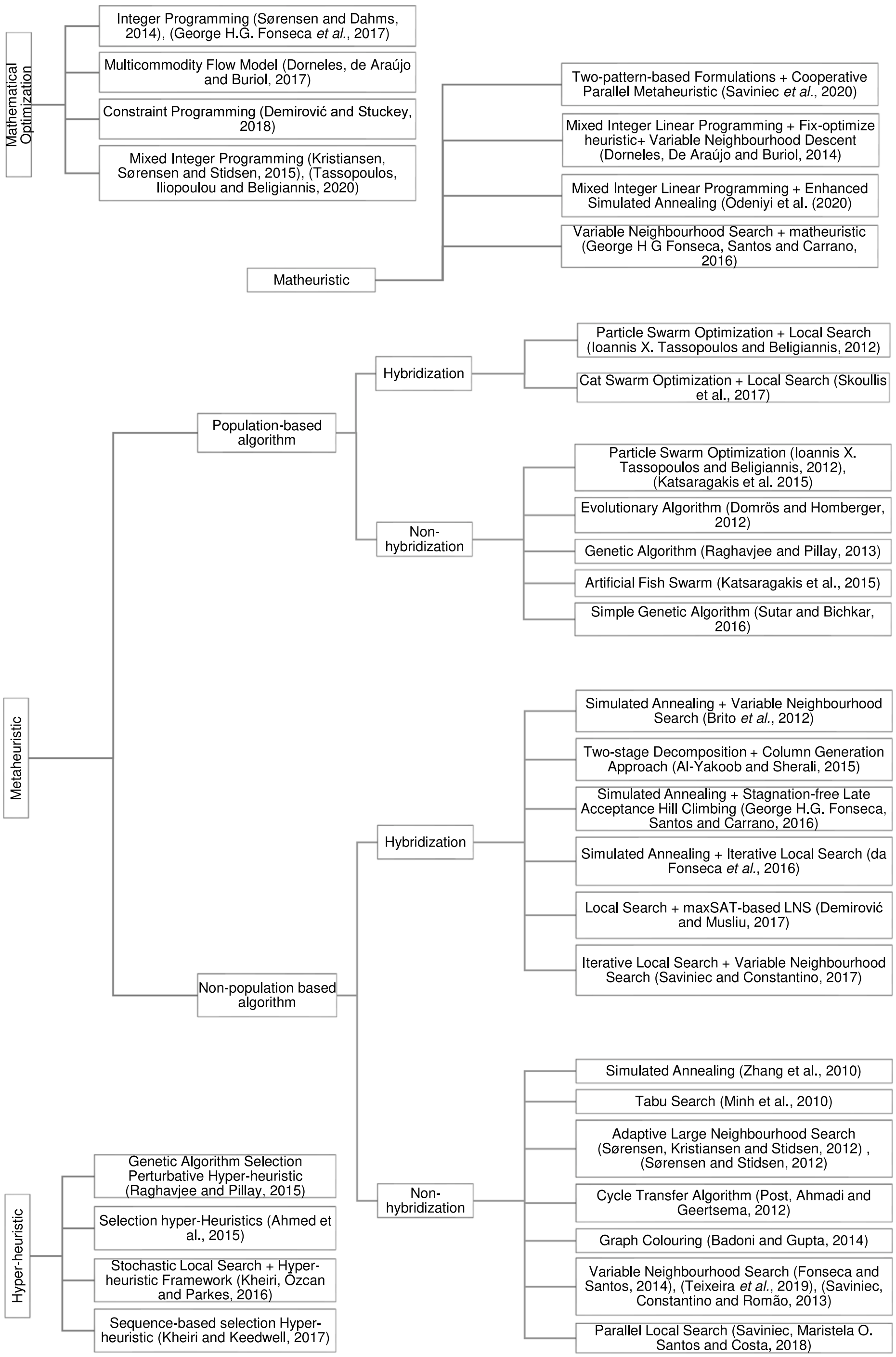
Mathematical Optimization
Integer Programming (Sørensen and Dahms, 2014), (George H.G. Fonseca et al., 2017)
Multicommodity Flow Model (Dorneles, de Araújo and Buriol, 2017)
Constraint Programming (Demirović and Stuckey, 2018)
Mixed Integer Programming (Kristiansen, Sørensen and Stidsen, 2015), (Tassopoulos, Iliopoulou and Beligiannis, 2020)
Matheuristic
Two-pattern-based Formulations + Cooperative Parallel Metaheuristic (Saviniec et al., 2020)
Mixed Integer Linear Programming + Fix-optimize heuristic+ Variable Neighbourhood Descent (Dorneles, De Araújo and Buriol, 2014)
Mixed Integer Linear Programming + Enhanced Simulated Annealing (Odeniyi et al. (2020)
Variable Neighbourhood Search + matheuristic (George H G Fonseca, Santos and Carrano, 2016)
Metaheuristic
Population-based algorithm
Hybridization
Particle Swarm Optimization + Local Search (Ioannis X. Tassopoulos and Beligiannis, 2012)
Cat Swarm Optimization + Local Search (Skoullis et al., 2017)
Non-hybridization
Particle Swarm Optimization (Ioannis X. Tassopoulos and Beligiannis, 2012), (Katsaragakis et al. 2015)
Evolutionary Algorithm (Domrös and Homberger, 2012)
Genetic Algorithm (Raghavjee and Pillay, 2013)
Artificial Fish Swarm (Katsaragakis et al., 2015)
Simple Genetic Algorithm (Sutar and Bichkar, 2016)
Non-population based algorithm
Hybridization
Simulated Annealing + Variable Neighbourhood Search (Brito et al., 2012)
Two-stage Decomposition + Column Generation Approach (Al-Yakoob and Sherali, 2015)
Simulated Annealing + Stagnation-free Late Acceptance Hill Climbing (George H.G. Fonseca, Santos and Carrano, 2016)
Simulated Annealing + Iterative Local Search (da Fonseca et al., 2016)
Local Search + maxSAT-based LNS (Demirović and Musliu, 2017)
Iterative Local Search + Variable Neighbourhood Search (Saviniec and Constantino, 2017)
Non-hybridization
Simulated Annealing (Zhang et al., 2010)
Tabu Search (Minh et al., 2010)
Adaptive Large Neighbourhood Search (Sørensen, Kristiansen and Stidsen, 2012) , (Sørensen and Stidsen, 2012)
Cycle Transfer Algorithm (Post, Ahmadi and Geertsema, 2012)
Graph Colouring (Badoni and Gupta, 2014)
Variable Neighbourhood Search (Fonseca and Santos, 2014), (Teixeira et al., 2019), (Saviniec, Constantino and Romão, 2013)
Parallel Local Search (Saviniec, Maristela O. Santos and Costa, 2018)
Hyper-heuristic
Genetic Algorithm Selection Perturbative Hyper-heuristic (Raghavjee and Pillay, 2015)
Selection hyper-Heuristics (Ahmed et al., 2015)
Stochastic Local Search + Hyper-heuristic Framework (Kheiri, Özcan and Parkes, 2016)
Sequence-based selection Hyper-heuristic (Kheiri and Keedwell, 2017)

Fig. 1. The categorisation of the solution methods for high school timetabling problems.

CreateNeighborhood, CalculateLocalCentre, Leap and Turbulence. Both algorithms were hybridised with a local search procedure to improve the quality of the timetable after the execution of the main algorithm. The hybrid PSO produced better results compared to the hybrid AFS.

The GOAL team proposed a hybridised Simulated Annealing (SA) and stagnation-free Late Acceptance Hill Climbing (sf-LAHC) to address ITC2011 instances (Fonseca et al., 2016a). Late Acceptance Hill Climbing (LAHC) was adapted from the HC method with the acceptance criteria of accepting the worse solution. The proposed sf-LAHC can reheat the system during stagnation, by retrieving the vector of costs from when the last improvement occurred. Overall, the search was performed by SA after an initial solution was generated by the KHE solver. sf-LAHC was further employed to improve the solution. Their hybrid algorithm outperformed the winner method by producing best-known solutions for 14 out of 18 instances.

Demirović and Musliu (2017) proposed a local search combined with a novel maxSAT-based large neighbourhood search in solving XHSTT formatted data. The Partial Weighted maxSAT formulation was used to model an XHSTT instance. Two operators (destroy and insertion operations) were used in the large neighbourhood search algorithm (LNS). An initial solution was refined by the local search algorithm; then the maxSAT-based LNS was used to further improve the solution. The hybrid algorithm was able to model 27 out of 39 instances.

Table 2
Features of the XHSTT formatted dataset.

Instance	Time	Teacher	Room	Class	Student	Event	Country
AU-BG-98	40	56	45	30	–	387	Australia
AU-SA-96	60	43	36	20	–	296	Australia
AU-TE-99	30	37	26	13	–	308	Australia
BrazilInstance 1	25	8	–	3	–	21	Brazil
BR-SA-00	25	14	–	6	–	63	Brazil
BrazilInstance 3	25	16	–	8	–	69	Brazil
BR-SM-00	25	23	–	12	–	127	Brazil
BrazilInstance 5	25	31	–	13	–	119	Brazil
BR-SN-00	25	30	–	14	–	140	Brazil
BrazilInstance 7	25	33	–	20	–	205	Brazil
CzechVillageSchool	30	8	–	6	–	67	Czech Republic
DK-FG-12	50	90	69	–	279	1077	Denmark
DK-HG-12	50	100	71	–	523	1235	Denmark
SmallSchool	4	9	–	–	74	25	Denmark
DK-VG-09	60	46	53	–	163	918	Denmark
ES-SS-08	35	66	4	21	–	225	Spain
Artificial school	20	22	12	13	–	169	Finland
FI-PB-98	40	46	34	31	–	387	Finland
ElementarySchool	35	22	21	60	–	291	Finland
FI-WP-06	35	18	13	10	–	172	Finland
FI-MP-06	35	25	25	14	–	280	Finland
SecondarySchool2	40	22	21	36	–	469	Finland
FirstHighSchoolAigio2010	35	37	–	208	–	283	Greece
GR-H1-97	35	29	–	66	–	372	Greece
GR-P3-10	35	29	–	84	–	178	Greece
ThirdHighSchoolPreveza2008	35	29	–	68	–	164	Greece
WesternGreeceUniversity3	35	19	–	6	–	210	Greece
GR-PA-08	35	19	–	12	–	262	Greece
WesternGreeceUniversity5	35	18	–	6	–	184	Greece
ItalyInstance 1	36	13	–	3	–	42	Italy
IT-I4-96	36	61	–	38	–	748	Italy
KS-PR-11	62	101	–	63	–	809	Kosovo
GEPRO	44	132	80	44	846	2675	Netherlands
NL-KP-03	38	75	41	18	453	1156	Netherlands
NL-KP-05	37	78	42	26	498	1235	Netherlands
Kottenpark2008	40	81	11	34	–	1027	Netherlands
NL-KP-09	38	93	53	48	–	1148	Netherlands
UK-SP-06	25	68	67	67	–	1227	England
US-WS-09	100	134	108	–	–	628	USA
ZA-LW-09	148	19	2	16	–	185	South Africa
ZA-WD-09	42	40	–	30	–	278	South Africa

From the 27 instances, the average solution cost is better than that of IP (Kristiansen et al., 2015).

Saviniec and Constantino (2017) proposed a hybridised Iterative Local Search (ILS) and Variable Neighbourhood Search (VNS) to solve Brazilian high school timetabling problems. Their implementation incorporated two neighbourhood structures (matching and torque operators) and local search routines to perform an effective search. The matching operator was used to create the best subset by finding the best local permutation of the required given class, whereas the torque operator was used to identifying the possible conflict of meetings by building a graph to pinpoint the connected components. Their method produced 38 best solutions out of 41 instances.

Skoullis et al. (2017) presented a hybrid Cat Swarm optimisation (CSO) with a local search algorithm in addressing the Greece high school timetabling problems. CSO was based on food seeking process of cats, where it consisted of a seek mode and a trace mode. In the seek mode, the environment were analysed to determine the next move when each cat was resting. In the trace mode, each cat moved rapidly to hunt for food. After 5000 iterations, a local search algorithm was executed to further refine the solution by minimising teachers' gaps. Their method exhibited better performance compared to a genetic algorithm, evolutionary algorithm, simulated annealing, particle swarm optimisation and artificial fish swarm.

4. A categorisation of the solution methods

Several state-of-the-art solution methods have been surveyed in this paper. The most recent solution methods for high school timetabling problems (irrespective of datasets) are presented in Table 1. We observe that the shift of popularity from local search methods to mathematical optimisation methods such as integer programming and constraint programming. Among the recent implementations include an IP (Tassopoulos et al., 2020) and a combination of meta-heuristic and mathematical optimisation (Saviniec et al., 2020).

The categorisation of the solution methods is given in Fig. 1. There are twenty-four meta-heuristics, eight mathematical optimisation methods, four hyper-heuristics and three matheuristics (the combination of meta-heuristic and mathematical optimisation technique). There are more application of meta-heuristics to the high school timetabling problems than the total of mathematical optimisation methods, hyper-heuristics and matheuristics combined. The majority of the meta-heuristics are non-population based methods. One drawback of using the population-based approach is the long execution time in finding a good quality solution (Kohshori & Abadeh, 2012; Sabar et al., 2013, 2014a). Variable neighbourhood search and adaptive large neighbourhood search seem to be popular choices in addressing the high school timetabling problems.

The mathematical optimisation techniques with an advanced formulation are capable of finding good/optimal solutions, especially for small problem instances (Fonseca et al., 2017; Sabar et al., 2015). In addition, the integer programming based methods show promising results on the Lectio dataset (Sørensen & Dahms, 2014), XHSTT formatted datasets (Fonseca et al., 2017), Brazilian high school dataset (Saviniec et al., 2020) and Greek high school dataset (Tassopoulos et al., 2020).

Meta-heuristic algorithms can achieve good quality solutions in a short time, but they do not ensure optimality (Fonseca et al., 2017;

Table 3
The features of public instances of the Brazilian high school dataset (Souza et al., 2003).

Instance	C	T	D	H	U	E	DE
1	3	8	5	5	40	75	21
2	6	14	5	5	25	150	29
3	8	16	5	5	80	200	4
4	12	23	5	5	170	300	41
5	13	31	5	5	0	325	71
6	14	30	5	5	10	350	63
7	20	33	5	5	0	500	84

Table 4
The features of real-case instances of the Brazilian high school dataset (Saviniec & Constantino, 2017).

Instance	C	T	D	H	U	E	DE
CM-CEUP-2011-N	3	15	5	5	284	75	36
FA-EEF-2011-M	4	12	5	5	160	100	42
JNS-CEJXXIII-2011-N	4	15	5	5	12	100	48
CM-CEDB-2010-N	5	17	5	5	41	125	60
JNS-CEJXXIII-2011-M	5	18	5	5	50	125	60
JNS-CEJXXIII-2011-V	5	18	5	5	52	125	60
JNS-CEDPII-2011-V	7	21	5	5	101	175	73
CM-CECM-2011-N	8	30	5	5	489	200	96
JNS-CEDPII-2011-M	8	19	5	5	91	200	85
CL-CECL-2011-N-A	9	28	5	5	25	225	107
MGA-CEVB-2011-V	9	20	5	5	214	225	97
MGA-CEVB-2011-M	10	21	5	5	167	250	108
CL-CEASD-2008-V-A	12	27	5	5	108	300	132
CL-CEASD-2008-V-B	12	27	5	5	108	300	132
MGA-CEDC-2011-V	12	31	5	5	412	300	131
CL-CECL-2011-M-A	13	31	5	5	23	325	144
CL-CECL-2011-M-B	13	31	5	5	8	325	143
CM-CECM-2011-V	13	34	5	5	455	325	142
CL-CECL-2011-V-A	14	29	5	5	21	350	164
CM-CEUP-2008-V	16	35	5	5	345	400	192
CM-CEUP-2011-M	16	38	5	5	498	400	192
CM-CEUP-2011-V	16	34	5	5	382	400	169
MGA-CEJXXIII-2010-V	16	35	5	5	309	400	192
NE-CESVP-2011-V-A	16	44	5	5	181	400	183
NE-CESVP-2011-V-B	16	43	5	5	192	400	184
NE-CESVP-2011-V-C	16	43	5	5	218	400	182
NE-CESVP-2011-M-A	18	45	5	5	156	450	212
NE-CESVP-2011-M-B	18	44	5	5	167	450	212
NE-CESVP-2011-M-C	18	45	5	5	152	450	211
NE-CESVP-2011-M-D	18	45	5	5	267	450	211
MGA-CEDC-2011-M	19	37	5	5	382	475	210
CM-CECM-2011-M	20	51	5	5	648	500	234
MGA-CEGV-2011-M	31	62	5	5	588	775	352
MGA-CEGV-2011-V	32	75	5	5	857	800	357

Sabar et al., 2013). On the other hand, exact methods are capable of obtaining optimal solutions but are less effective for medium and large size problems (Dorneles et al., 2017, 2014; Saviniec et al., 2020). Therefore, researchers integrated both exact (mathematical optimisation) method and meta-heuristic in an approach called matheuristic.

For the XHSTT formatted dataset, KHE is often used to generate initial solutions. KHE is a platform specifically built for XHSTT formatted dataset (Kingston, 2014). It works by allocating difficult lessons first. It is effective in producing good quality solutions.

5. Data format of school timetabling problem

The different data formats for high school timetabling around the world makes algorithm comparison difficult if not impossible. A standard data format is required to achieve this aim. Burke et al. (1998) proposed a standard data format for timetabling to overcome the difficulty of data exchange. The proposed format is like the Z specification language.

Table 5
The features of selected instances from lectio dataset (Sorensen & Stidsen, 2013).

Instance	E	EC	R	D	M	T	A	C
AalborTG2012	768	296	116	5	7	35	231	202
AarhusA2011	768	50	101	5	6	30	383	306
AarhusA2012	803	44	94	5	6	30	437	302
Aars2009	682	10	47	5	9	45	158	209
Aars2010	1112	480	46	10	9	90	143	154
Aars2011	924	286	46	10	7	70	140	154
Aars2012	713	99	46	10	6	60	116	141
Alssund2010	729	293	38	5	9	45	214	229
Alssund2012	1473	3	34	10	9	90	215	245
BagsvaG2010	278	28	33	5	6	30	102	95
BirkerG2011	1637	193	81	5	9	45	587	439
BirkerG2012	1574	250	79	5	9	45	102	461
BjerrG2009	730	16	33	5	9	45	197	313
BjerrG2010	545	186	35	5	9	45	208	160
BjerrG2011	571	195	42	5	9	45	202	175
BjerrG2012	564	180	42	5	9	45	208	175
BroendG2012	389	38	31	10	4	40	74	102
CPHWGym2010	467	196	39	5	9	45	225	147
CPHWGym2011	511	223	39	5	9	45	245	165
CPHWGym2012	525	218	41	5	9	45	285	169
CPHWHG2012	634	6	27	5	8	40	217	163
CPHWHTX2010	530	28	35	5	10	50	111	124
CPHWHTX2011	688	213	34	5	10	50	110	121
CPHWHTX2012	434	13	30	5	10	50	89	82
DetFG2012	858	220	51	5	11	55	343	167
DetKG2010	185	8	26	5	7	35	111	69
DetKG2011	195	10	26	5	7	35	111	76
EUCN2009	249	18	55	5	7	35	89	78
EUCN2010	583	173	55	5	7	35	192	189
EUCN2011	286	108	64	5	7	35	29	98
EUCN2012	303	72	64	5	7	35	107	100
EUCNHG2010	190	83	29	5	8	40	49	56
EUCS2012	252	81	19	5	9	45	50	59
FaaborgG2008	1754	0	47	10	14	140	188	180
FalkonG2009	1139	2	71	10	5	50	468	326
FalkonG2011	955	1	72	10	5	50	369	284
FalkonG2012	1120	17	68	10	5	50	369	313
GUAasia2010	555	10	36	5	12	60	33	141
GUQaqor2011	524	262	15	10	10	100	90	70
GUQaqor2012	518	259	15	10	10	100	73	61
HadersK2011	662	3	75	5	5	25	483	318
HasserG2010	1275	75	71	10	5	50	526	384
HasserG2011	1369	86	79	10	5	50	585	408
HasserG2012	1471	146	70	10	5	50	623	422
HerningG2010	135	25	88	5	6	30	7	5
HerningG2011	1783	79	97	10	6	60	269	363
HerningG2012	1924	87	107	10	6	60	276	411
HoejeTaG2008	201	0	74	5	8	40	99	66
HoejeTaG2009	610	0	74	5	8	40	254	207
HoejeTaG2010	607	0	74	5	8	40	228	202
HoejeTaG2011	688	0	76	5	8	40	267	226
HoejeTaG2012	827	12	76	5	8	40	270	268
HorsenS2009	380	1	50	5	5	25	297	195
HorsenS2012	1119	5	54	10	5	50	551	409
Johann2012	1304	249	67	5	8	40	202	419
KalundG2011	1701	177	64	10	7	70	376	281
KalundG2012	1654	198	66	10	7	70	425	299
KalundHG2010	376	44	17	5	9	45	87	98
KoebenPG2012	95	1	22	5	6	30	63	43
KoegeH2012	1092	425	64	5	9	45	214	294
KongshoG2010	441	5	69	5	4	20	301	245
MariageG2009	692	10	71	10	4	40	240	183
MorsoeG2012	584	41	40	10	5	50	237	161

(*continued on next page*)

Reis and Oliveira (2001) proposed a comprehensive UniLang format for data representation applicable to school, university and examination timetabling problems. A simple translator tool was developed to translate the problems into the standard format.

Özcan (2005) proposed a new Extensible Markup Language (XML) data format for the timetabling problem by utilising MathML (an XML based standard for describing mathematical expressions) content

Table 5 (*continued*).

Instance	E	EC	R	D	M	T	A	C
NaerumG2008	1533	0	75	10	4	40	567	513
NaerumG2009	1435	0	77	10	4	40	93	483
NielsSG2011	265	0	73	5	10	50	96	74
NielsSG2012	669	174	73	5	10	50	111	235
NordfynG2012	771	51	60	10	4	40	239	209
NyborgG2011	1191	4	59	10	5	50	466	336
OdderCfU2010	766	15	73	5	11	55	467	231
OdderG2009	782	2	47	10	6	60	175	199
OdderG2012	843	22	41	10	5	50	184	219
OrdrupG2010	1044	521	52	10	8	80	151	133
OrdrupG2011	1564	782	52	10	8	80	236	229
RibeK2011	837	2	61	5	8	40	263	227
RysenG2010	1477	36	74	10	4	40	396	319
RysenG2011	1294	26	74	10	4	40	427	395
RysenG2012	1382	59	75	10	4	40	469	516
SanktAG2012	773	16	47	10	4	40	63	210
SkanderG2010	1116	11	57	10	6	60	69	277
SkanderG2011	1161	14	56	10	6	60	463	284
SkanderG2012	1275	25	57	10	6	60	571	289
SkiveG2010	2665	1241	58	10	9	90	304	331
SlagelG2012	2152	164	103	10	6	60	607	469
SoendS2011	1206	76	98	10	4	40	383	332
SoendS2012	1278	1	111	10	4	40	340	379
StruerS2012	2915	160	71	10	9	90	348	401
VardeG2012	887	1	65	10	5	50	251	277
VejenG2009	928	10	52	10	6	60	209	189
Vejlefjo2011	714	63	45	5	14	70	186	234
VestfynG2009	585	234	64	5	8	40	260	168
VestfynG2010	582	239	62	5	8	40	246	167
VestfynG2011	619	255	57	5	8	40	257	180
VestfynG2012	613	254	51	5	8	40	195	180
ViborgK2011	1302	4	59	10	6	60	456	308
ViborgTG2009	549	90	27	5	10	50	87	120
ViborgTG2010	454	6	21	10	5	50	86	110
ViborgTG2011	473	42	23	10	4	40	90	109
VirumG2012	1731	225	65	10	4	40	536	443
VordingbG2009	615	57	67	5	7	35	262	227

Table 6
The features of instances of Greek high school dataset.

Instance	No. of teachers	No. of classes	No. of teaching hours per week
1	34	11	35
2	35	11	35
3	19	6	35
4	19	7	35
5	18	6	35
6	34	10	35
7	35	13	35
8	11	5	30
9	15	6	{32, 34, 35}
10	17	7	30
11	21	9	{33, 34, 35}

markup, named Timetabling Markup Language (TTML). A TTML translator called CONFETI, was able to convert a final exam timetabling data into TTML format under some conditions.

XML archive for High School Timetabling (XHSTT) format was used in the Third International Timetabling Competition (ITC2011) (Post et al., 2016). XHSTT is composed of four entities, namely times, resources, events and constraints (Post, Ahmadi, Daskalaki et al., 2012; Post et al., 2014). It is the most widely used format by researchers in school timetabling (Pillay, 2014).

5.1. Benchmark dataset

In this section, we give an overview of the popular datasets used by researchers. Fig. 2 shows the solution methods for high school timetabling problems, according to datasets.

5.1.1. XHSTT formatted dataset

This dataset can be downloaded from the University of Twente website.[1] To date, a total of 41 instances from 13 different countries is provided in the archive (see Table 2). An HSEval validator[2] is included for solution verification. For this dataset, non-population based approach is widely used compared to population-based algorithm. Simulated Annealing (SA) is commonly hybridised with other meta-heuristics. This is possibly due to its good performance in the past. Variable Neighbourhood Search is another popular methodology. Integer Programming (Fonseca et al., 2017; Kristiansen et al., 2015) and matheuristics (Dorneles et al., 2014; Fonseca et al., 2016b) are the current state-of-the-art methods for this dataset.

5.1.2. Brazilian high school dataset

This dataset consists of 41 instances, of which seven are public instances (Souza et al., 2003) and 34 are real-case instances from 13 high schools in Brazil (Saviniec & Constantino, 2017). The features of the instances are displayed in C = number of classes; T = number of teachers; D = number of days; H = periods per days; U = total number of periods for which the teachers are unavailable; E = total number of lessons; and DE = total number of required double lessons (Tables 3 and 4). Six different methodologies have been proposed for this dataset. The parallel local search algorithm (Saviniec et al., 2018) is superior to the other five in terms of performance.

5.1.3. Lectio dataset

This dataset can be accessed through the online high school administration system Lectio by the Danish high school administrative staffs (Sørensen & Stidsen, 2012). It contains thousands of instances from hundreds of different high schools (Sorensen & Stidsen, 2013). However, only up to a selection of 100 instances were used in the literature (Sørensen & Dahms, 2014). Table 5 shows the features of the dataset. Note that E = events; EC = number of event chains; R = rooms; D = days; M = modules; T = timeslots; A = set of entities (comprised of students and teachers); and C = classes. For the Lectio high school dataset, two different algorithms were used. The integer programming approach (Sørensen & Dahms, 2014) was more effective than adaptive large neighbourhood search.

5.1.4. Greek high school dataset

This dataset consists of eleven instances and can be obtained from the University Of Patras.[3] All the instances are made from real-world high school timetabling problems by Tassopoulos and Beligiannis (2012b) and Tassopoulos et al. (2020). Table 6 shows the features of the dataset. Nine methodologies were proposed. Mixed Integer Programming (Tassopoulos et al., 2020) outperformed all the population-based algorithms.

5.1.5. OR-library dataset

The dataset can be downloaded from OR-library webpage.[4] The features of the five artificial high school instances used in the scientific literature (Raghavjee & Pillay, 2013, 2015) are listed in Table 7. They are hard timetabling problems as all periods must be utilised with very little or no options for each allocation (Sutar & Bichkar, 2016). For the OR library dataset, three population-based algorithms and one non-population-based algorithm were applied to five "hard" artificially generated high school timetabling problems. For the population-based algorithms, two were meta-heuristics (two-phase genetic algorithm and simple genetic algorithm) and another was a hyper-heuristic (genetic algorithm selection perturbative hyper-heuristic). For the non-population-based algorithm is a simulated annealing algorithm. All the algorithms managed to find feasible solutions.

[1] https://www.utwente.nl/en/eemcs/dmmp/hstt/.

[2] http://www.it.usyd.edu.au/jeff/cgi-bin/hseval.cgi.

[3] http://www.deapt.upatras.gr/school_timetabling_MIP_modeling/.

[4] http://people.brunel.ac.uk/mastjjb/jeb/orlib/tableinfo.html.

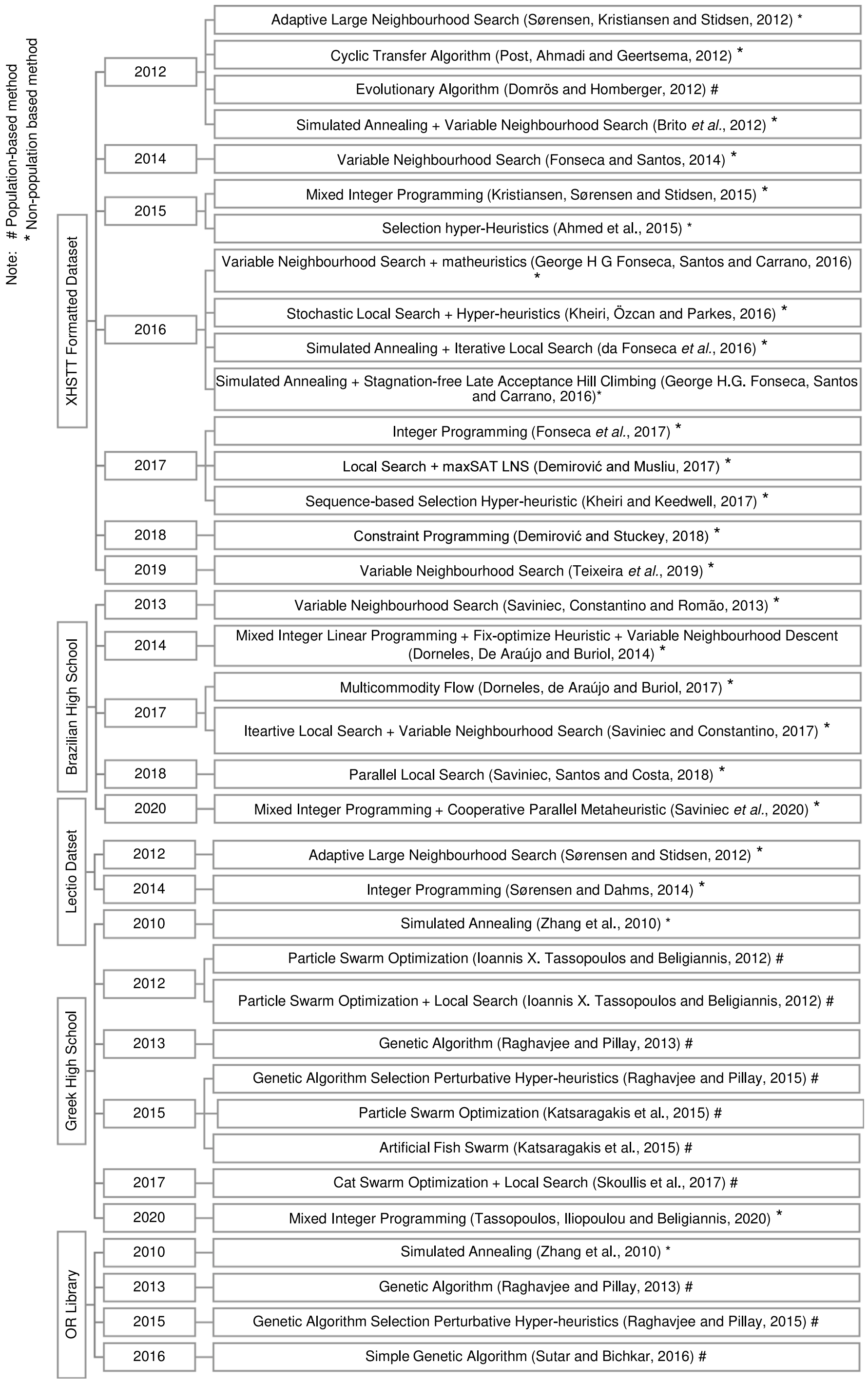
Note: # Population-based method
* Non-population based method
XHSTT Formatted Dataset
2012
Adaptive Large Neighbourhood Search (Sørensen, Kristiansen and Stidsen, 2012) *
Cyclic Transfer Algorithm (Post, Ahmadi and Geertsema, 2012) *
Evolutionary Algorithm (Domrös and Homberger, 2012) #
Simulated Annealing + Variable Neighbourhood Search (Brito et al., 2012) *
2014
Variable Neighbourhood Search (Fonseca and Santos, 2014) *
2015
Mixed Integer Programming (Kristiansen, Sørensen and Stidsen, 2015) *
Selection hyper-Heuristics (Ahmed et al., 2015) *
2016
Variable Neighbourhood Search + matheuristics (George H G Fonseca, Santos and Carrano, 2016) *
Stochastic Local Search + Hyper-heuristics (Kheiri, Özcan and Parkes, 2016) *
Simulated Annealing + Iterative Local Search (da Fonseca et al., 2016) *
Simulated Annealing + Stagnation-free Late Acceptance Hill Climbing (George H.G. Fonseca, Santos and Carrano, 2016)*
2017
Integer Programming (Fonseca et al., 2017) *
Local Search + maxSAT LNS (Demirović and Musliu, 2017) *
Sequence-based Selection Hyper-heuristic (Kheiri and Keedwell, 2017) *
2018
Constraint Programming (Demirović and Stuckey, 2018) *
2019
Variable Neighbourhood Search (Teixeira et al., 2019) *
Brazilian High School
2013
Variable Neighbourhood Search (Saviniec, Constantino and Romão, 2013) *
2014
Mixed Integer Linear Programming + Fix-optimize Heuristic + Variable Neighbourhood Descent (Dorneles, De Araújo and Buriol, 2014) *
2017
Multicommodity Flow (Dorneles, de Araújo and Buriol, 2017) *
Iteartive Local Search + Variable Neighbourhood Search (Saviniec and Constantino, 2017) *
2018
Parallel Local Search (Saviniec, Santos and Costa, 2018) *
2020
Mixed Integer Programming + Cooperative Parallel Metaheuristic (Saviniec et al., 2020) *
Lectio Datset
2012
Adaptive Large Neighbourhood Search (Sørensen and Stidsen, 2012) *
2014
Integer Programming (Sørensen and Dahms, 2014) *
Greek High School
2010
Simulated Annealing (Zhang et al., 2010) *
2012
Particle Swarm Optimization (Ioannis X. Tassopoulos and Beligiannis, 2012) #
Particle Swarm Optimization + Local Search (Ioannis X. Tassopoulos and Beligiannis, 2012) #
2013
Genetic Algorithm (Raghavjee and Pillay, 2013) #
2015
Genetic Algorithm Selection Perturbative Hyper-heuristics (Raghavjee and Pillay, 2015) #
Particle Swarm Optimization (Katsaragakis et al., 2015) #
Artificial Fish Swarm (Katsaragakis et al., 2015) #
2017
Cat Swarm Optimization + Local Search (Skoullis et al., 2017) #
2020
Mixed Integer Programming (Tassopoulos, Iliopoulou and Beligiannis, 2020) *
OR Library
2010
Simulated Annealing (Zhang et al., 2010) *
2013
Genetic Algorithm (Raghavjee and Pillay, 2013) #
2015
Genetic Algorithm Selection Perturbative Hyper-heuristics (Raghavjee and Pillay, 2015) #
2016
Simple Genetic Algorithm (Sutar and Bichkar, 2016) #

Fig. 2. The solution methods for high school timetabling problems according to datasets.

Table 7
Features of the selected five instances from OR-library datasets.

Instance	No. of teachers	No. of venues	No. of classes	No. of periods	No. of tuples
hdtt4	4	4	4	30	120
hdtt5	5	5	5	30	130
hdtt6	6	6	6	30	180
hdtt7	7	7	7	30	210
hdtt8	8	8	8	30	240

Table 8
Software packages for school timetable construction.

Package	Country
Action Timetable Software	South Africa
ascTimetables	Slovakia
Docendo	Denmark
EduSwift	India
EduSys	India
Edutimer	India
Edval	Australia
Fedena	India
Free Timetable Software	Germany
Hashmicro Timetable Generator	Singapore
Mimosa	Finland
Primary Time	Australia
Prime Timetable	USA
Skolaris	Czech Republic
Timetable Generator	India
TimeTabler	United Kingdom
Timetabler Plus	Malaysia and Singapore
Untis Express	Austria

6. Industry perspective

There are several software packages developed for school timetable construction. A few of them are held in the cloud and can be used without the need for installation. The primary concern of this software is to avoid clashing of resources. An additional constraint may include room preferences, subject distribution, the idle period for teachers, teacher workload limits, teacher preferences, catering for class, teacher unavailabilities etc. Generally, the software is embedded with interactive management tools so that it can be easily utilised by school administrative staffs without intensive training. Their focus is on ease of use instead of developing sophisticated optimisation techniques in constructing school timetables. Table 8 shows a list of commercial software and freeware for school timetable construction and their country of origin.

7. Future direction

There is a variety of school timetabling datasets available (Section 5) and a fair amount of methodologies have been developed for these datasets (Section 3). XHSTT dataset is a more popular option compared to other datasets among the researchers. This dataset is publicly available online. It has drawn the attention of many researchers after ITC2011. In addition, this dataset is relatively new and more challenging in terms of the number of constraints involved compared to other datasets.

From observation, most algorithms used non-population based instead of population-based methodologies. As evident in the scientific literature, non-population based algorithms seem more promising compared to population-based algorithm. In fact, the winner of ITC2011 used a non-population-based algorithm (da Fonseca et al., 2016). The Integer Programming based method is the current state-of-the-art for XHSTT (Fonseca et al., 2017), Lectio (Sørensen & Dahms, 2014) and Greek (Tassopoulos et al., 2020) datasets. This indicates the potential of IP based method in producing good quality solutions. From the literature, mathematical formulation plays an important role in determining the success of IP methods (Burke et al., 2005).

The software industry appears to focus on user experience, whereas researchers from academia tend to focus on the development of sophisticated optimisation techniques. The collaboration between these two parties is required in order to develop a user-friendly yet effective school timetabling software that would benefit the community.

8. Conclusion

The research in high school timetabling has not advanced in a rate comparable to other educational timetabling such as university course timetabling and examination timetabling. This paper presents an overview of different methodologies employed in high school timetabling problems. The methodologies are grouped according to years, techniques and datasets. In addition, this paper provides details on high school timetabling datasets. The industry perspective and a list of software packages for school timetable construction are presented. Lastly, this paper gives possible future directions for research in this domain. We encourage the collaboration between industry and academia to develop an effective and easy to use school timetabling software.

CRediT authorship contribution statement

Joo Siang Tan: Conceptualization, Investigation, Visualisation, Writing - original draft. **Say Leng Goh:** Supervision, Conceptualization, Methodology, Resources, Validation, Writing - review & editing. **Graham Kendall:** Methodology, Conceptualization, Writing - review & editing. **Nasser R. Sabar:** Methodology, Conceptualization, Formal analysis, Writing - review & editing.

Declaration of competing interest

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

References

Ahmed, L. N., Özcan, E., & Kheiri, A. (2015). Solving high school timetabling problems worldwide using selection hyper-heuristics. *Expert Systems with Applications*, *42*, 5463–5471.

Akkan, C., & Gulcu, A. A bi-criteria hybrid Genetic Algorithm with robustness objective for the course timetabling problem. *90*, 22–32.

Al-Yakoob, S. M., & Sherali, H. D. (2015). Mathematical models and algorithms for a high school timetabling problem. *Computers & Operations Research*, *61*, 56–68.

Andrade, C. E., Silva, T., & Pessoa, L. S. (2019). Minimizing flowtime in a flowshop scheduling problem with a biased random-key genetic algorithm. *Expert Systems with Applications*, *128*, 67–80.

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

Badoni, R. P., & Gupta, D. K. (2014). A graph edge colouring approach for school timetabling problems. *International Journal of Mathematics in Operational Research*, *6*(1), 123–138.

Brito, S. S., Fonseca, G. H., Toffolo, T. A., Santos, H. G., & Souza, M. J. (2012). A SA-VNS approach for the high school timetabling problem. *Electronic Notes in Discrete Mathematics*, *39*, 169–176.

Burke, E. K., & Kendall, G. (2005). *Search methodologies*. Springer.

Burke, E. K., Kingston, J. H., & Pepper, P. A. (1998). A standard data format for timetabling instances. In *Computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*: *vol. 1408*, (pp. 213–222).

da Fonseca, G. H. G., Santos, H. G., Toffolo, T. Â. M., Brito, S. S., & Souza, M. J. F. (2016). GOAL solver: a hybrid local search based solver for high school timetabling. *Annals of Operations Research*, *239*(1), 77–97.

Demirović, E., & Musliu, N. (2017). Maxsat-based large neighborhood search for high school timetabling. *Computers & Operations Research*, *78*, 172–180.

Demirović, E. ., & Stuckey, P. J. (2018). Constraint programming for high school timetabling: A scheduling-based model with hot starts. In *International conference on the integration of constraint programming, artificial intelligence, and operations research* (pp. 135–152).

Deris, S., Omatu, S., & Ohta, H. (2000). Timetable planning using the constraint-based reasoning. *27*, 819–840.

Domrös, J., & Homberger, J. (2012). An evolutionary algorithm for high school timetabling. In *Proceedings of the ninth international conference on the practice and theory of automated timetabling (PATAT 2012) (August)* (pp. 485–488).

Dorneles, Á. P., de Araújo, O. C., & Buriol, L. S. (2017). A column generation approach to high school timetabling modeled as a multicommodity flow problem. *European Journal of Operational Research*, *256*(3), 685–695.

Dorneles, Á. P., De Araújo, O. C., & Buriol, L. S. (2014). A fix-and-optimize heuristic for the high school timetabling problem. *Computers & Operations Research*, *52*(PART A), 29–38.

Dutta, D., Sil, J., & Dutta, P. (2020). A bi-phased multi-objective genetic algorithm based classifier. *Expert Systems with Applications*, Article 113163.

Fonseca, G. H., & Santos, H. G. (2014). Variable neighborhood search based algorithms for high school timetabling. *Computers & Operations Research*, *52*, 203–208.

Fonseca, G. H., Santos, H. G., & Carrano, E. G. (2016a). Late acceptance hill-climbing for high school timetabling. *Journal of Scheduling*, *19*(4), 453–465.

Fonseca, G. H., Santos, H. G., & Carrano, E. G. (2016b). Integrating matheuristics and metaheuristics for timetabling. *Computers & Operations Research*, *74*, 108–117.

Fonseca, G. H., Santos, H. G., Carrano, E. G., & Stidsen, T. J. (2017). Integer programming techniques for educational timetabling. *European Journal of Operational Research*, *262*(1), 28–39.

Goh, S. L., Kendall, G., & Sabar, N. R. (2019). Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem. *The Journal of the Operational Research Society*, *70*(6), 873–888.

Katsaragakis, I., Tassopoulos, I., & Beligiannis, G. (2015). A comparative study of modern heuristics on the school timetabling problem. *Algorithms*, *8*(3), 723–742.

Kheiri, A., & Keedwell, E. (2017). A hidden markov model approach to the problem of heuristic selection in hyper-heuristics with a case study in high school timetabling problems. *Evolutionary Computation*, *25*(5), 473–501.

Kheiri, A., Özcan, E., & Parkes, A. J. (2016). A stochastic local search algorithm with adaptive acceptance for high-school timetabling. *Annals of Operations Research*, *239*(1), 135–151.

Kingston, J. H. (2014). KHE14: An algorithm for high school timetabling. In *Proceedings of the tenth international conference on practice and theory of automated timetabling (August 2014)* (pp. 269–291).

Kohshori, M. S., & Abadeh, M. S. (2012). Hybrid genetic algorithms for university course timetabling. *International Journal of Computer Science Issues*, *9*(2), 446–455.

Kristiansen, S., Sørensen, M., & Stidsen, T. R. (2015). Integer programming for the generalized high school timetabling problem. *Journal of Scheduling*, *18*(4), 377–392.

Kristiansen, S., & Stidsen, T. R. (2013). *A comprehensive study of educational timetabling, a survey, Vol. 978-87-931 (November)* (p. 72).

Leite, N., Melício, F., & Rosa, A. C. (2019). A fast simulated annealing algorithm for the examination timetabling problem. *Expert Systems with Applications*, *122*, 137–151.

Lewis, R. (2008). A survey of metaheuristic-based techniques for university timetabling problems. *OR Spectrum*, *30*(1), 167–190.

Minh, K. N. T. T., Thanh, N. D. T., Trang, K. T., & Hue, N. T. T. (2010). Using tabu search for solving a high school timetabling problem. In *Advances in intelligent information and database systems* (pp. 305–313).

Odeniyi, O. A., Omidiora, E. O., Olabiyisi, S. O., & Oyeleye, C. A. (2020). A mathematical programming model and enhanced simulated annealing algorithm for the school timetabling problem. (pp. 21–38).

Özcan, E. (2005). Towards an XML-based standard for timetabling problems: TTML. *Multidisciplinary Scheduling: Theory and Applications*, 163–185.

Pillay, N. (2014). A survey of school timetabling research. *Annals of Operations Research*, *218*(1), 261–293.

Post, G., Ahmadi, S., Daskalaki, S., Kingston, J. H., Kyngas, J., Nurmi, C., & Ranson, D. (2012). An XML format for benchmarks in high school timetabling. *Annals of Operations Research*, *194*(1), 385–397.

Post, G., Ahmadi, S., & Geertsema, F. (2012). Cyclic transfers in school timetabling. *OR Spectrum*, *34*(1), 133–154.

Post, G., Di Gaspero, L., Kingston, J. H., McCollum, B., & Schaerf, A. (2016). The third international timetabling competition. *Annals of Operations Research*, *239*(1), 69–75.

Post, G., Kingston, J. H., Ahmadi, S., Daskalaki, S., Gogos, C., Kyngas, J., Nurmi, C., Musliu, N., Pillay, N., Santos, H., & Schaerf, A. (2014). XHSTT: an XML archive for high school timetabling problems in different countries. *Annals of Operations Research*, *218*(1), 295–301.

Qu, R., Burke, E. K., McCollum, B., Merlot, L. T. G., & Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, *12*(1), 55–89.

Raghavjee, R., & Pillay, N. (2013). A study of genetic algorithms to solve the school timetabling problem. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*: *vol. 8266 LNAI*, (pp. 64–80). (PART 2).

Raghavjee, R., & Pillay, N. (2015). A genetic algorithm selection perturbative hyper-heuristic for solving the school timetabling problem. *ORiON*, *31*(1), 39.

Reis, L. P., & Oliveira, E. (2001). A language for specifying complete timetabling problems. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*: *vol. 2079 LNCS*, (pp. 322–341).

Sabar, N. R., Ayob, M., Kendall, G., & Qu, R. (2013). Grammatical evolution hyper-heuristic for combinatorial optimization problems. *IEEE Transactions on Evolutionary Computation*, *17*(6), 840–861.

Sabar, N. R., Ayob, M., Kendall, G., & Qu, R. (2014a). Automatic design of a hyper-heuristic framework with gene expression programming for combinatorial optimization problems. *IEEE Transactions on Evolutionary Computation*, *19*(3), 309–325.

Sabar, N. R., Ayob, M., Kendall, G., & Qu, R. (2014b). A dynamic multiarmed bandit-gene expression programming hyper-heuristic for combinatorial optimization problems. *IEEE Transactions on Cybernetics*, *45*(2), 217–228.

Sabar, N. R., Ayob, M., Qu, R., & Kendall, G. (2012). A graph coloring constructive hyper-heuristic for examination timetabling problems. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *37*(1), 1–11.

Sabar, N. R., Zhang, X. J., & Song, A. (2015). A math-hyper-heuristic approach for large-scale vehicle routing problems with time windows. In *2015 IEEE congress on evolutionary computation (CEC)* (pp. 830–837). IEEE.

Saviniec, L., & Constantino, A. A. (2017). Effective local search algorithms for high school timetabling problems. *Applied Soft Computing Journal*, *60*, 363–373.

Saviniec, L., Constantino, A. A., & Romão, W. (2013). Vns based algorithms to the high school timetabling problem. In *Anais do XLV simpósio brasileiro de pesquisa operacional* (pp. 845–856).

Saviniec, L., Santos, M. O., & Costa, A. M. (2018). Parallel local search algorithms for high school timetabling problems. *European Journal of Operational Research*, *265*(1), 81–98.

Saviniec, L., Santos, M. O., Costa, A. M., & Santos, L. M. (2020). Pattern-based models and a cooperative parallel metaheuristic for high school timetabling problems. *European Journal of Operational Research*, *280*(3), 1064–1081.

Schaerf, A. (1999a). Local search techniques for large high school timetabling problems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *29*(4), 368–377.

Schaerf, A. (1999b). Survey of automated timetabling. *Artificial Intelligence Review*, *13*(2), 87–127.

Skoullis, V. I., Tassopoulos, I. X., & Beligiannis, G. N. (2017). Solving the high school timetabling problem using a hybrid cat swarm optimization based algorithm. *Applied Soft Computing*, *52*, 277–289.

Sørensen, M., & Dahms, F. H. (2014). A two-stage decomposition of high school timetabling applied to cases in Denmark. *Computers & Operations Research*, *43*(1), 36–49.

Sørensen, M., Kristiansen, S., & Stidsen, T. R. (2012). International timetabling competition 2011: An adaptive large neighborhood search algorithm. In *9th international conference on the practice and theory of automated timetabling (PATAT 2012), (August)* (pp. 489–492).

Sørensen, M., & Stidsen, T. R. (2012). High school timetabling: Modeling and solving a large number of cases in Denmark. In *Proceedings of the ninth international conference on the practice and theory of automated timetabling (PATAT 2012)* (pp. 359–364).

Sorensen, M., & Stidsen, T. J. R. (2013). *Comparing solutions approches for a complete model of high school timetabling*: *DTU Management Engineering Report n. 5.2013*, (pp. 1–45).

Souza, M. J. F., Maculan, N., & Ochi, L. S. (2003). A GRASP-tabu search algorithm for solving school timetabling problems. (pp. 659–672).

Sutar, S. R., & Bichkar, R. S. (2016). Genetic algorithms based timetabling using knowledge augmented operators. *International Journal of Computer Science and Information Security*, *14*(11), 570.

Tassopoulos, I. X., & Beligiannis, G. N. (2012a). A hybrid particle swarm optimization based algorithm for high school timetabling problems. *Applied Soft Computing*, *12*(11), 3472–3489.

Tassopoulos, I. X., & Beligiannis, G. N. (2012b). Solving effectively the school timetabling problem using particle swarm optimization. *Expert Systems with Applications*, *39*(5), 6029–6040.

Tassopoulos, I. X., Iliopoulou, C. A., & Beligiannis, G. N. (2020). Solving the greek school timetabling problem by a mixed integer programming model. *The Journal of the Operational Research Society*, *71*(1), 117–132.

Teixeira, U. R., Souza, M. J. F., de Souza, S. R., & Coelho, V. N. (2019). An adaptive VNS and skewed GVNS approaches for school timetabling problems. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*: *vol. 11328 LNCS*, (pp. 101–113).

Woumans, G., De Boeck, L., Beliën, J., & Creemers, S. (2016). A column generation approach for solving the examination-timetabling problem. *European Journal of Operational Research*, *253*(1), 178–194.

Yuan, S., Li, T., & Wang, B. (2020). A co-evolutionary genetic algorithm for the two-machine flow shop group scheduling problem with job-related blocking and transportation times. *Expert Systems with Applications*, Article 113360.

Zhang, D., Liu, Y., M'Hallah, R., & Leung, S. C. H. (2010). A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. *European Journal of Operational Research*, *203*(3), 550–558.

Zheng, S., Wang, L., Liu, Y., & Zhang, R. (2015). A simulated annealing algorithm for university course timetabling considering travelling distances. *International Journal of Computing Science and Mathematics*, *6*(2), 139–151.