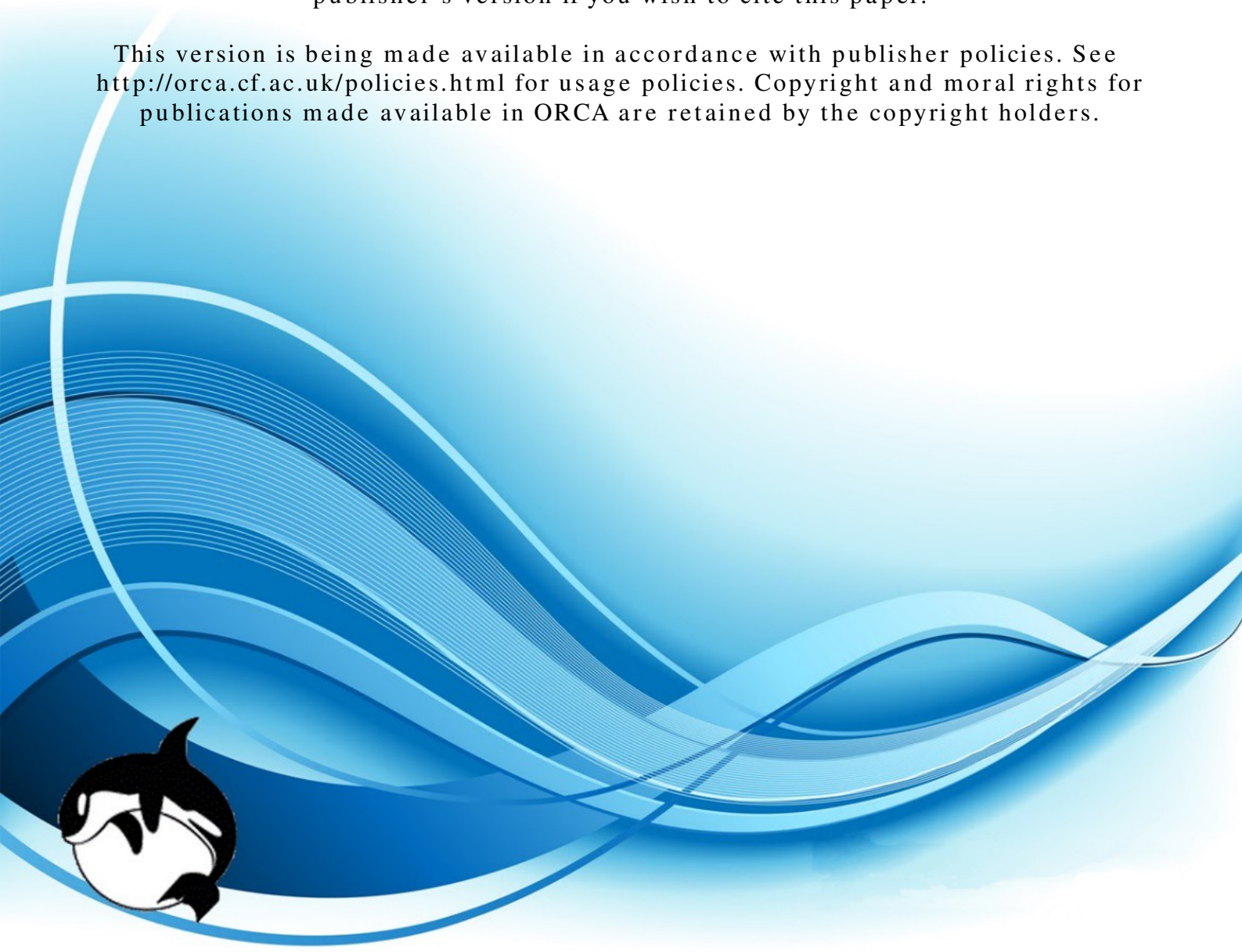


Annals of Operations Research manuscript No.
(will be inserted by the editor)

A Stochastic Local Search Algorithm with Adaptive Acceptance for High-school Timetabling

Ahmed Kheiri · Ender Özcan · Andrew J. Parkes

Received: date / Accepted: date

Abstract Automating high school timetabling is a challenging task. This problem is a well known hard computational problem which has been of interest to practitioners as well as researchers. High schools need to timetable their regular activities once per year, or even more frequently. The exact solvers might fail to find a solution for a given instance of the problem. A selection hyper-heuristic can be defined as an easy-to-implement, easy-to-maintain and effective 'heuristic to choose heuristics' to solve such computationally hard problems. This paper describes the approach of the team HySST (Hyper-heuristic Search Strategies and Timetabling) to high school timetabling which competed in all three rounds of the Third International Timetabling Competition. HySST generated the best new solutions for three given instances in Round 1 and gained the second place in Rounds 2 and 3. It achieved this by using a fairly standard stochastic search method but significantly enhanced by a selection hyper-heuristic with an adaptive acceptance mechanism.

Keywords Timetabling · Stochastic Local Search · Hyper-heuristic · Restart · Scheduling

1 Introduction

Due to the inherent difficulties in educational timetabling, this area of study has been of interest to many researchers and practitioners across operational research, computer science and artificial intelligence since 1960s (Broder, 1964). There are different types of educational timetabling problems, such as examination timetabling and high school timetabling (Pillay, 2010a) which are all known to be NP-hard real-world constraint optimisation problems (Even et al, 1976; de Werra, 1997). The last competition in the series of timetabling challenges was recently organised on high school timetabling: the Third International Timetabling Competition (ITC2011). ITC2011 aimed at providing a high school timetabling bench-

University of Nottingham, School of Computer Science
Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, UK
{ axk, exo, ajp }@cs.nott.ac.uk

mark, determining the state-of-the-art solution method and promoting researchers and practitioners to deal with the problems as they are without any simplification. This study describes the approach of the HySST (Hyper-heuristic Search Strategies and Timetabling) team which competed in the three rounds of ITC2011.

Many different tailored methodologies have been proposed for solving specific high school timetabling problems. We describe a fairly standard stochastic search method but significantly enhanced by a *selection hyper-heuristic* for high school timetabling. The proposed approach is more general in the sense that it is applicable to a variety of high school timetabling problems across the world and more effective than the existing approaches which frequently ignore the real-world complexities. A *hyper-heuristic* is a high level search methodology that performs search over the space formed by a set of low level heuristics to solve computationally hard problems (Burke et al, 2013). A *selection hyper-heuristic* improves an initially generated solution iteratively through *heuristic selection* and *move acceptance* processes (Özcan et al, 2008). A candidate solution is perturbed after applying a chosen heuristic using the heuristic selection method and a new solution is obtained. Then a move acceptance method considers whether to accept or reject the new solution. This cycle continues until a set of termination criteria is satisfied. There have been a growing number of studies on hyper-heuristics since the initial ideas of combining the strength of existing heuristics (neighbourhood structures) was rooted in 1960s (Fisher and Thompson, 1963). More on different types of hyper-heuristics including selection hyper-heuristics can be found in (Chakhlevitch and Cowling, 2008; Ross, 2005; Burke et al, 2013). A selection hyper-heuristic should be "fast to implement, requiring far less expertise in either the problem domain or heuristic methods, and robust enough to effectively handle a range of problems" (Cowling et al, 2001). HySST generated the best new solutions for three given instances in Round 1 and gained the second place in Rounds 2 and 3 with the proposed selection hyper-heuristic, and which also satisfies the above design criteria.

Section 2 discusses high school timetabling and selection hyper-heuristic frameworks. Section 3 provides an overview of the competition and high school timetabling problem dealt with during the competition. Section 4 describes the selection hyper-heuristic components that are used for solving the high school timetabling problem. Section 5 presents the empirical and so competition results. Finally, Section 6 concludes the study.

2 Background

2.1 High School Timetabling Problem

High school timetabling (HST) is a well-known NP-hard real-world combinatorial optimisation problem (Even et al, 1976; de Werra, 1997). A solution requires scheduling of events such as courses, classes and resources such as teachers, rooms and more using a number of time slots subject to a set of constraints. The constraints are classified as *hard* and *soft*. The solutions respecting all hard constraints are considered *feasible* and they are expected to satisfy as many of the soft constraints as possible, which represent preferences. In most of the previous formulations of the high school timetabling problem, infeasible solutions are allowed and

evaluated, differentiating their quality by considering the degree of hard constraint violations.

There are a variety of real world high school timetabling problems exhibiting various characteristics from different countries and many different approaches have been proposed for a particular problem in hand, including a tiling algorithm (Kingston, 2005), constraint programming approach (Valouxis and Housos, 2003; Marte, 2007), Hopfield neural networks (Smith et al, 2003) and integer programming (Birbas et al, 2009). The tabu search or simulated annealing metaheuristics are frequently preferred as the single-point-based solution methods for high school timetabling. Abramson (1991) employed simulated annealing for course timetabling and proposed a parallel algorithm for solving some randomly generated problem instances and some Australian data. Hertz (1992) utilised tabu search for teacher-course assignment using hypothetical data from a Yugoslavian school. Schaerf (1996) tested a tabu search based approach which interleaves different types of moves on some instances from the Italian high-schools. The approach generated schedules that are of better quality than the manually created ones. Abramson et al (1999) compared different simulated annealing cooling schedules and the experimental results showed that geometric cooling with multiple rates performs the best. Jacobsen et al (2006) presented a tabu search algorithm for solving a timetabling problem at German secondary schools of Gymnasium type and compared its performance to a constraint programming approach. The results showed that they have a similar performance based on the feasible solutions obtained for the given instances. Bello et al (2008) tested a tabu search approach on some instances that are presentative of Brazilians high school timetabling problems. Kannan et al (2012) applied graph theoretic approach to a problem from the New York City public school system, which decomposes a given instance and applies randomised heuristics.

The number of studies investigating into population-based metaheuristics, particularly hybrid evolutionary algorithms for high school course timetabling has been growing since the 1990s (Ross et al, 1994; Corne et al, 1994; Erben and Keppler, 1996). Colorni et al (1992) compared various metaheuristics based on GA, simulated annealing and tabu search using an Italian high-school data. Their results indicate that GAs hybridised with local search is promising. Filho et al (2001) formulated the timetabling problem as a clustering problem and applied a genetic algorithm to construct solutions to the timetabling problem of public schools in Brazil. Wilke et al (2002) proposed a hybrid genetic algorithm using multiple genetic operators and a parameter configuration strategy that randomly chooses from different options during the search process whenever the algorithm detects that no improvement can be made. The results showed that the proposed hybrid approach performed better than the traditional genetic algorithm on a large German high school problem instance. Beligiannis et al (2008) presented an evolutionary algorithm which employs no crossover and multiple mutation operators. A comparison to the previously proposed approaches of column generation and constraint programming on a Greek school course timetabling problem revealed the success of the approach. Raghavjee and Pillay (2008) compared the performance of a genetic algorithm, neural network, simulated annealing, tabu search and greedy search on the problem instances provided by Abramson and Dang (1993). The experimental results showed that genetic algorithm delivered either a better or similar performance to the previously proposed methods. Raghavjee and Pillay

(2010) described a hybrid evolutionary algorithm with no crossover using a hill climber for solving a South African high school course timetabling problem along with a primary school timetabling problem. Moura and Scaraficci (2010) applied the greedy randomised adaptive search procedure (GRASP) heuristic to solve a Brazilian high school timetabling problem. Pillay (2010b) implemented an evolutionary algorithm based hyper-heuristic selection method. The study revealed that the incorporation of local search heuristics with mutation and crossover operators improves the performance. The approach outperforms the other methods applied to the same problem. Özcan et al (2012) introduced a variant of a high school timetabling problem from Turkey and proposed a genetic algorithm hybridised with hill climbing which interleaves the proposed algorithm with constructive methods while exploiting the underlying hierarchical structure of a given problem. More on high school timetabling can be found in (Pillay, 2010a, 2012). This study presents a selection hyper-heuristic method which is able to solve a variety of high school timetabling problems from different countries.

2.2 Selection Hyper-heuristic Frameworks

Figure 1(a) illustrates how a high level generic selection hyper-heuristic operates. A selection hyper-heuristic manages a set of *perturbative* or *constructive* low level heuristics (move operators) (Burke et al, 2010b) and often improves an initially generated solution (s_i) under an iterative framework until the termination criterion is satisfied. We focus on the former type of selection hyper-heuristics. Özcan et al (2008) identified two successive stages that are common to most of the single-point-based search hyper-heuristics influencing their performances: heuristic selection and move acceptance. Most of the simple selection hyper-heuristics are introduced by Cowling et al (2001). For example, A *simple random* heuristic selection method chooses a random low level heuristic at each step, while *random permutation* produces a random permutation of all low level heuristics and applies the low level heuristic in the list one after another at each step. *Greedy* applies all low level heuristics to a given solution and selects the best heuristic which produces the best solution (which could be worse than the given solution, if all heuristics are performing random perturbation). Selection hyper-heuristics have been successfully applied to many different real world problems, including channel assignment (Kendall and Mohamad, 2004), examination timetabling (Özcan et al, 2009), space allocation (Burke et al, 2005), vehicle routing problems (Pisinger and Ropke, 2007). More on hyper-heuristics including the descriptions of more elaborate selection hyper-heuristic components as well as other types of hyper-heuristics can be found in Burke et al (2013).

A *mutational* operator in the context of search randomly perturbs a given solution and the new solution is not guaranteed to be of the same quality as the given one, whereas a *hill climbing* operator always returns a non-worsening solution which might be the same as the given solution. A generic selection hyper-heuristic does not differentiate between the types of low level heuristics. Özcan et al (2008) proposed different types of selection hyper-heuristic frameworks showing that a generalised version of iterated local search (Lourenço et al, 2010) performed well on some benchmark functions which separates mutational and hill climbing operators, invoking them successively and so enforcing diversification and intensification

processes explicitly. Burke et al (2010a) also showed that a similar hyper-heuristic framework performs well on a hyper-heuristic benchmark[1]. In this study, we use a multistage approach as shown in Figure 1(b), which separates mutational and hill climbing heuristics as well. Mutational heuristics are employed until some criteria are satisfied which decides that it is time for intensification, and then a new stage starts employing only hill climbing low level heuristics. The proposed framework allows switching back and forth between diversification and intensification stages.

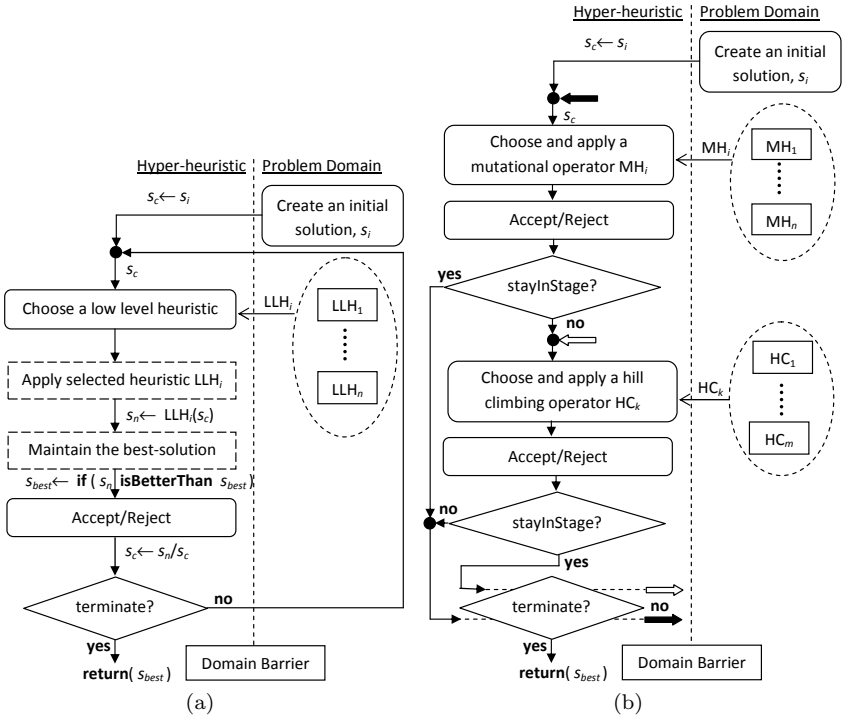


Fig. 1 Illustration of a (a) generic and (b) multistage selection hyper-heuristic framework

The idea of reducing the number of low level heuristics within selection hyper-heuristics has been studied previously. Cowling and Chakhlevitch (2003) tested *peckish* heuristic selection strategies within hyper-heuristics on a real-world personnel scheduling problem. Although those strategies are found effective, selecting a low-level heuristic to apply at each decision point turned out to be slow since a large set of low level heuristics were used. Hence, Chakhlevitch and Cowling (2005) investigated learning strategies for choosing the subset of low-level heuristics with *good* performance. At each step, the total improvement due to a heuristic is updated and used as a measure to choose that subset. The approach which linearly reduces the number of the good performing low level heuristics in the subset shown to be promising. Mustafa (2012) introduced an adaptive strategy within

[1] www.asap.cs.nott.ac.uk/chesc2011

a selection hyper-heuristic which identifies poorly performing low level heuristics and discards them during the search process. Özcan and Kheiri (2011) proposed a multistage hyper-heuristic in which the number of low level heuristics combined with their parametric choices are reduced based on the trade-off between improvement achieved by a low level heuristic with a given setting and the time it takes to achieve that performance. The selection hyper-heuristic methods described above do not make use of the nature of the low level heuristics while selecting and applying them to the candidate solution. In contrast, our approach to high school timetabling uses the type of the heuristic. It is based on a selection hyper-heuristic managing a reduced set of low level heuristics, with the method employing either only mutational or only hill climbing low level heuristics at a given stage. Moreover, the proposed framework enables the use of two separate heuristic selection mechanisms at each stage.

3 The Third International Timetabling Competition (ITC2011)

Due to the variety of existing high school timetabling problems and sometimes lack of algorithmic details, it is not trivial to implement and compare the performance of different approaches. More importantly, many previous studies frequently focus on simplified models of high school timetabling because of its inherent difficulty. The International Timetabling Competitions have been organised with the goal of encouraging researchers and practitioners to design solution methods for real world problems incorporating all real world complexities into their models and form real world benchmark for the timetabling community. The state-of-the-art methods for a given domain has always been of interest for researchers as well as practitioners, which has been the case for timetabling as well. The Third International Timetabling Competition (ITC2011)[2] was recently organised after ITC2002 (http://www.idsia.ch/Files/ttcomp2002/) and ITC2007 (McCollum et al, 2010) which were on educational timetabling, mainly focusing on university course and examination timetabling.

The focus of ITC2011 was high school timetabling. The competition consisted of three rounds. In the first round, competitors were invited to submit solutions to all public instances with the goal of finding the best approach that improves upon the best known solution for each instance. In the second round, a time limit was imposed as 1000 nominal seconds based on the organisers' computer. For each of the hidden instances, ten runs with different random seeds were conducted considering submission of stochastic algorithms. The solutions obtained from each run for each instance were ranked and then averaged to determine the winner. In the third round, the hidden instances were published and the competitors were invited to submit the best solutions that they can achieve by any algorithm. The same ranking strategy as the second round was used during this round to determine the winner. The high school timetabling instances were obtained across the world based on different education systems, where each problem came with its particular format. A unified format was required. Post et al (2012) proposed and used a common XML data format to represent a given problem instance of ITC2011 as input.

[2] ITC2011 website: http://www.utwente.nl/ctit/hstt/

3.1 Problem definition

The ITC2011 problem instances (Post et al, 2012) contain four components including *times*, *resources*, *events* and *constraints*. A time component represents an indivisible interval of time during which an event runs. A resource represents the entity which attends an event. For example, teacher, room, student or class are resources. An event is a meeting between resources. A constraint is the condition that a solution must/should satisfy, if possible. The ITC2011 problem instances contain 15 types of constraints overall: *assign resource, assign time, split events, distribute split events, prefer resources, prefer times, avoid split assignments, spread events, link events, avoid clashes, avoid unavailable times, limit idle times, cluster busy times, limit busy times, limit workload* (Post et al, 2012). In a standard fashion, constraints are separated into *hard* and *soft*. Each constraint has a boolean variable called *Required* to indicate whether the constraint is hard or soft. In the ITC2011 competition, such constraints are not strictly hard but are simply much more heavily penalised than the 'soft' constraints.

A candidate solution is evaluated in terms of two components: *feasibility* and *preferences*. The evaluation function computes the weighted hard and soft constraint violations for a given solution, where the weights are pre-defined in the input file representing a given instance, as *infeasibility* and *objective* values, respectively. The quality of a solution is denoted concatenating those two values as in $infeasibility - value.objective - value$ using sufficient number of digits in the objective-value part and filling with 0s if necessary. For example, 10.000090 represents an infeasibility value of 10 and objective value of 90. For the comparison of algorithms, a solution is considered to be better than another one, if it has a smaller infeasibility value, or an equal infeasibility value and a smaller objective value. Post et al (2012) provides a more detailed description of the high school timetabling problem and ITC2011.

3.2 ITC2011 Dataset

As a total of twenty one high school timetabling problem instances were made public during the first round of the competition. Eighteen hidden instances were used during the second round of the competition which are then made public and used for the third round of the competition. Table 1 summarises the main characteristics of all problem instances obtained from different countries. These characteristics give some rough idea about the size of each instance, yet do not define a given problem fully as the importance of violating a given constraint is not provided. The ITC2011 dataset can be downloaded from the competition website.

4 A Multistage Hyper-heuristic Search for High School Timetabling

A stochastic local search algorithm is implemented for solving ITC2011 high school timetabling problems, based on the selection hyper-heuristic framework as described in section 2.2 (Figure 1(b)), and as a time contract algorithm which terminates after a given time, $t_{overall}$ for each instance. The approach consists of an initial solution construction phase followed by an extensive improvement phase using

Table 1 Characteristics of the problem instances used during the three rounds of the competition

Round 1						
Instance - Country	Times	Teachers	Rooms	Classes	Students	Duration
BGHS98 - Australia	40	56	45	30		1564
SAHS96 - Australia	60	43	36	20		1876
TES99 - Australia	30	37	26	13		806
Instance1 - Brazil	25	8		3		75
Instance5 - Brazil	25	31		13		325
Instance7 - Brazil	25	33		20		500
StPaul - England	27	68	67	67		1227
ArtificialSchool - Finland	20	22	12	13		200
College - Finland	40	46	34	31		854
HighSchool - Finland	35	18	13	10		297
SecondarySchool - Finland	35	25	25	14		306
HighSchool1 - Greece	35	29		66		372
Patras 3rd HS 2010 - Greece	35	29		84		340
Preveza 3rd HS 2008 - Greece	35	29		68		340
Instance1 - Italy	36	13		3		133
GEPRO - Netherlands	44	132	80	44	846	2675
Kottenpark2005 - Netherlands	37	78	42	26	498	1272
Lewitt2009 - South Africa	148	19	2	16		838
Common to All Rounds						
Instance4 - Brazil	25	23		12		300
Instance6 - Brazil	25	30		14		350
Kottenpark2003 - Netherlands	38	75	41	18	453	1203
Rounds 2 and 3						
Instance2 - Brazil	25	14		6		150
Instance3 - Brazil	25	16		8		200
ElementarySchool - Finland	35	22	21	60		445
SecondarySchool2 - Finland	40	22	21	36		566
Aigio 1st HS 2010 - Greece	35	37		208		532
Instance4 - Italy	36	61		38		1101
Instance1 - Kosovo	62	101		63		1912
Kottenpark2005A - Netherlands	37	78	42	26	498	1272
Kottenpark2008 - Netherlands	40	81	11	34		1118
Kottenpark2009 - Netherlands	38	93	53	48		1301
Woodlands2009 - South Africa	42	40		30		1353
School - Spain	35	66	4	21		439
WesternGreeceUni3 - Greece	35	19		6		210
WesternGreeceUni4 - Greece	35	19		12		262
WesternGreeceUni5 - Greece	35	18		6		184

a multistage selection hyper-heuristic. The pseudocode of the algorithm is provided in Figure 1. The initial construction of a complete solution is performed using the general solver implemented by Jeff Kingston as the KHE library[3]. Note that the construction phase often yields a solution in which many constraints are still violated requiring further enhancement. The improvement phase uses the remaining time left ($t_{remaining}$) after the construction of the initial solution which takes t_{init} time. Until the given time limit is reached, the proposed approach switches between a diversification stage (stage A) which employs a selection hyper-heuristic combining simple random heuristic selection with an adaptive move acceptance and an intensification stage (stage B) which employs a strict hill climbing process based on two heuristics (see section 2.2). Each stage takes a prefixed amount of time ($t_{MUstage}$ and $t_{HCstage}$). Moreover, stage A controls a threshold value ϵ to relax the degree of consecutive worsening moves during the search process.

[3] `http://sydney.edu.au/engineering/it/~jeff/khe/`

Algorithm 1 Pseudocode of the proposed multistage hyper-heuristic.

```
1: procedure HySST_Solver_ITC2011( t_overall, t_MUstage, t_HCstage )
2:     S ← create_initial_solution();                          ▷ takes t_init time
3:     t_remaining ← t_overall − t_init;
4:     S_best ← S;
5:     thresholdList[] = {ϵ_1, ϵ_2, ..., ϵ_maxLevel};
6:     level ← 1;
7:     while t_remaining notExceeded do
8:         S_stage_best ← S;
9:         S_stage_start ← S;
10:        ϵ ← thresholdList[level];                           ▷ stage A entry using ϵ
11:        while t_MUstage notExceeded && t_remaining notExceeded do
12:            LLH ← SelectRandomlyFrom(MutationalHeuristics);
13:            S′ ← ApplyHeuristic(LLH, S);
14:            if S′ isBetterThan S_best then
15:                S_best ← S′;
16:            end if
17:            if S′ isBetterThan S_stage_best then
18:                S_stage_best ← S′;
19:            end if
20:            S = MoveAcceptance(S, S′, S_stage_best, ϵ);   ▷ threshold acceptance
21:        end while
22:        if S_stage_best isNotBetterThan S_stage_start then   ▷ stage B entry
23:            while t_HCstage notExceeded && t_remaining notExceeded do
24:                LLH ← SelectRandomlyFrom(HillClimbers);
25:                S″ ← ApplyHeuristic(LLH, S);
26:                if S″ isBetterThan S_best then
27:                    S_best ← S″;
28:                end if
29:                if S″ isBetterThan S_stage_best then
30:                    S_stage_best ← S″;
31:                end if
32:                S ← S″;                                   ▷ accept all moves
33:            end while
34:        end if
35:        if S_stage_best isNotBetterThan S_stage_start then
36:            if level == maxLevel then
37:                S ← S_stage_start;
38:                level ← 1;
39:            else
40:                level + +;
41:            end if
42:        end if
43:    end while
44:    return S_best;
45: end procedure
```

The diversification stage makes use of eight mutational low level heuristics allowing worsening moves to be accepted via a naïve move acceptance method. The usefulness of restart in randomised search algorithms has already been known and different approaches have been proposed (Kautz et al, 2002; Luby et al, 1993). In this study, we use an adaptive threshold move acceptance method to enable acceptance of worsening moves and partial restarts. The threshold move acceptance method accepts all improved solutions or a worsening solution with a quality better than $(1 + \epsilon)$ of the quality of the best solution obtained during the search process at a stage. The acceptance of a worsening solution in this manner can be

considered as a partial restart on a given solution. The degree of a partial restart is indicated by *level* controlling the threshold value of ϵ. The larger the threshold is, the lower the quality of solutions that get accepted. The diversification stage is repeated within the time limits as long as the best solution obtained at the end of a stage (S_{stage_best}) is of better quality than the best solution in hand at the start of a stage (S_{stage_start}). In a way, the diversification stage is parametrised depending on ϵ. Each diversification stage using a different ϵ is considered as a different stage. If a diversification stage produces a worsening resultant solution, then the intensification stage which makes use of two hill climbing heuristics kicks in. If a solution cannot be improved even after an intensification stage, the ϵ value is increased to allow even larger changes in the solution causing larger worsening in its quality in the stage. We have used a discrete choice for the ϵ values and grabbed the next (previous) item from an ordered fixed-size list of threshold values in order to increase (decrease) its value. The minimum and maximum threshold values are limited using the first and last items in the list.

4.1 Low level heuristics

Two selection hyper-heuristics are employed operating cooperatively and mixing a set of 10 domain-specific low-level heuristics which are (mostly) fairly simple moves such as moving a task to a different resource, or swaps of events.

During the diversification-stage, the selection hyper-heuristic manages eight mutational move operators:

- **MH**$_1$ swaps the start time of two randomly selected events. For example, assuming that the $Mathematics$ class meeting is assigned to the first time slot on Monday and the $History$ class meeting assigned to the third time slot on Friday, after the swap operation, $History$ is assigned to the first time slot on Monday, while $Mathematics$ to the third time slot on Friday.
- **MH**$_2$ randomly selects an event and reschedules it to a random time. For example, assuming that the $Mathematics$ class meeting is assigned to the first time slot on Monday, after applying this heuristic, $Mathematics$ could be rescheduled to the last time slot on Friday.
- **MH**$_3$ swaps the time of two randomly chosen events. If both events have the same duration, this heuristic operates like MH$_1$, but if their durations are not the same then the first chosen event is moved to the time slot right after the second event ends. For example, when swapping a $Mathematics$ class meeting with a duration of one assigned to the first time slot on Friday with a $History$ class meeting with a duration of two assigned to the second time slot on Friday, MH$_3$ moves the $Mathematics$ class to the third time slot on Friday, rather than the second time slot, and moves the $History$ class to the first time slot on Friday.
- **MH**$_4$ selects a random resource element within an event and modifies its assignment randomly. For example, assuming that $Classroom1$ is assigned for the $Physics$ meeting, after applying this heuristic, $Classroom1$ can be reassigned for a meeting of $Mathematics$.
- **MH**$_5$ swaps two random resources. For example, assuming that $Classroom1$ is assigned for $Mathematics$ and $Classroom2$ is assigned for $History$, after

applying this heuristic, $Classroom1$ is assigned for the $History$ lesson while $Classroom2$ is assigned for the $Mathematics$ lesson.
- $\mathbf{MH}_6$ reassigns a randomly chosen resource element of an event to a random resource. For example, assuming that $Teacher1$ is assigned to teach $Mathematics$, after applying this heuristic, $Teacher1$ gets replaced by $Teacher8$.
- $\mathbf{MH}_7$ merges two class meetings of the same event and adjacent in time. For example, assuming that the *Biology* class meeting with a duration of two is assigned to the first time slot on Monday and another *Biology* class meeting with a duration of one is assigned to the third time slot on Monday, then after applying the heuristic, the *Biology* class meetings are merged into a single meeting with a duration of three starting at the first time slot on Monday.
- $\mathbf{MH}_8$ splits a randomly selected event requiring an assignment of a time block consisting of multiple time slots into two events with separate times with a fixed low probability of 0.1%. For example, assuming that a *Biology* class meeting is randomly chosen which has an assignment of a time block of two consecutive time slots, MH_8 divides the teaching of *Biology* into two separate (but still consecutive) time slots without changing their current assignments allowing future moves to operate on those two meetings separately.

During the intensification-stage, the selection hyper-heuristic mixes two hill climbing heuristics. Unlike most of the mutational operators, these two hill-climbing operators are capable of making quite large changes to a solution. The intensification-stage itself is slightly non-standard. One of the operators is designed using neighbourhood structures based on ejection chains, while the other one is a type of first improvement hill climbing operator. Both hill climbing operators attempt to make moves which respect a particular constraint type, while hoping to improve upon the other types of constraint violations, but might have a net worsening of the objective, however, then such worsening moves are rejected. A hill climbing step is always non-worsening and so can be repeatedly applied in a standard fashion until a local minimum is reached. A notable difference from standard methods (such as in memetic algorithms) is that we found that performance is better, if the hill climbing is not applied whenever the mutational operators managed to improve the best solution. We suspect that excessive use of the hill climbing somehow gives over-optimised local solutions that afterwards lead to restricted movement within the search space.

5 Results

We joined ITC2011 as the team HySST (Hyper-heuristic Search Strategies and Timetabling) using the approach described in Section 4. At the end of the competition, there were four additional teams who were able to submit solutions for ITC2011: GOAL, HFT, Lectio and VAGOS. In this section, we present the results.

5.1 Multistage improvement using adaptive move acceptance

The multistage stochastic local search hyper-heuristic managing all low level heuristics and using the adaptive move acceptance for partial restarts turned out to be very effective in solving high school timetabling problems. If a small value of ϵ

does not provide any improvement in the solution quality in stage A, then its value is increased which causes acceptance of lower quality solutions and escape from a local optimum. Figure 2 provides a sample run on Instance4-Brazil using $\epsilon = \{0.001, 0.33, 1.99\}$ ignoring the strict hill climbing process in stage B. The plot shows that the *reheats* do lead to drastic drops in the cost of a solution, and without the reheats, the search is clearly stuck. For example, the stage indicated as level 1 in Figure 2 (red points) performing almost strict hill climbing based on $\epsilon = 0.001$ is eventually stuck and even the stage indicated as level 2 in which ϵ is 0.33 (green points) gets stuck. The blue points (level 3) in Figure 2 are a strong relaxation where $\epsilon = 1.99$, but do lead to later improvements.

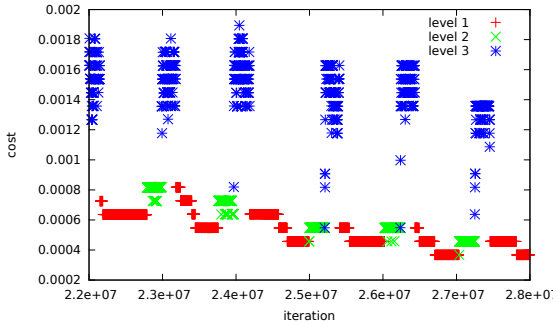


Fig. 2 Cost versus iteration plot of a sample run towards the end of the search process which is obtained by applying the proposed approach to Instance4-Brazil using the threshold list of {0.001, 0.33, 1.99}.

The hill climbing algorithms fail to produce an improving solution most of the time and for most of the instances. For example, Figure 3 displays a sample run where hill climbing yields no improvement in any stage. Yet, it has been observed that the stage B based on hill climbing is useful for achieving high quality solutions; in particular, for the Australian high school timetabling instances. At the end of a run on those instances, the proposed approach using hill climbing yield (even if slightly) better results than the one which does not use hill climbing. Figure 4 shows a sample run on the BGHS98 instance with and without hill climbing (stage B). After the mutational heuristics are employed at a stage A, regardless of the threshold level, hill climbing generates a non-worsening feasible solution. Unfortunately, this seems to occur once and no more improvement could be achieved via the hill climbing algorithms.

5.2 Competition results

The proposed hyper-heuristic successfully improved upon the best previously known solutions (BKNs) for the Australian high school timetabling instances of BGHS98, SAHS96 and TES99 in the first round of the competition as shown in Table 2.

Table 3 summarises the results of Round 2 on the hidden instances. The column labelled as "KHE" shows the average quality of ten initial solutions produced

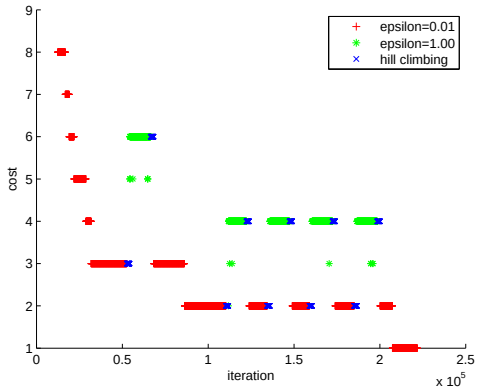
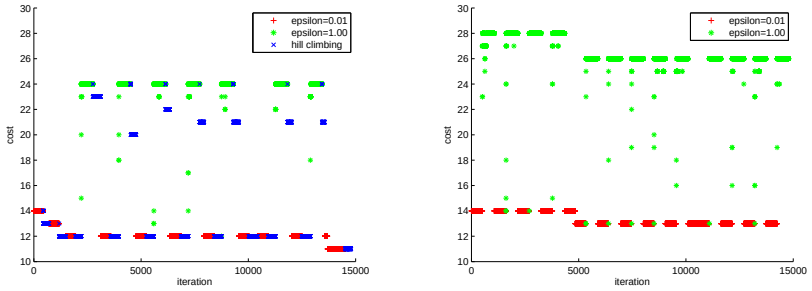


Fig. 3 Cost versus iteration plot of a sample run on WesternGreeceUni5.



(a) (b)

Fig. 4 Cost versus iteration plot for BGHS98 - Australia (a) with and (b) without hill climbing.

Table 2 The performance of the HySST approach in Round 1. The quality (cost) of a solution is indicated as feasibility-value.objective-value and BKN is the best previously known solution.

Dataset	BKN	HySST
BGHS98 - Australia	7.433	3.494
SAHS96 - Australia	23.044	8.052
TES99 - Australia	26.134	1.140

by the constructive approach of the KHE library. The best feasibility/objective values over ten runs for each instance show that HySST performs the best on two instances of Kottenpark2003 and Kottenpark2005A from Netherland and worst on Instance1 - Kosovo. The results reveal that HFT and Lectio did not use the default constructive approach and they obtained solutions of quality which are even worse than the constructive approach achieves for 16 and 6 instances, respectively. Since the GOAL team submitted the Brazilian timetabling instances they are not

considered for ranking for the first four instances. The table 3 provides, also, the average ranks of each approach based on their ranking for each instance per run. The proposed hyper-heuristic turns out to be the second best approach.

Table 3 The performance comparison of the HySST approach to the other competing approaches over 10 trials showing the best quality (cost) of a solution indicated as feasibility-value.objective-value in Round 2. The best values are highlighted in bold.

Problem	KHE	HySST	GOAL	HFT	Lectio
Instance2	3.20001	1.00069	1.00051	5.00183	**0.00019**
Instance3	3.50002	0.00096	**0.00087**	26.00264	0.00112
Instance4 - Brazil	39.10001	2.00238	16.00104	63.00225	**1.00172**
Instance6	11.60003	2.00229	4.00207	21.00423	**0.00183**
ElementarySchool	9.90000	0.00004	**0.00003**	29.00080	**0.00003**
SecondarySchool2	1.80017	0.00006	**0.00000**	28.01844	0.00014
Aigio 1st HS 2010	12.20008	0.00322	**0.00006**	45.03665	0.00653
Instance4 - Italy	32.60218	0.04012	**0.00169**	250.05966	0.00225
Instance1	1307.10005	1065.17431	**38.09789**	986.42437	274.04939
Kottenpark2003	4.40747	**0.47560**	0.87084	203.87920	34.55960
Kottenpark2005A	32.70292	**26.35251**	27.37026	393.40463	185.83973
Kottenpark2008	72.51725	32.71562	**10.33034**	INVALID	84.99999
Kottenpark2009	48.22637	33.99999	**25.14030**	337.99999	97.96060
Woodlands2009	13.90000	2.00047	2.00012	59.00336	**0.00094**
School	2.50039	0.01247	**0.00597**	63.13873	0.01927
WesternGreeceUni3	0.00024	0.00010	**0.00005**	14.00198	30.00002
WesternGreeceUni4	0.00044	0.00016	**0.00005**	233.00277	35.00070
WesternGreeceUni5	15.40000	0.00001	**0.00000**	9.00174	4.00013
Average ranking		2.23	**1.18**	3.64	2.32

Table 4 summarises the feasibility/objective values obtained by the five competitors' solvers including the developed hyper-heuristic (HySST) on the hidden instances on Round three. The proposed hyper-heuristic produced the best results in six including three ties out of eighteen instances. The GOAL team was not considered for the Brazilian instances for ranking in this round as well. The table 4 provides, also, the average ranks of each competing approach in round 3. Our selection hyper-heuristic became the second best approach.

6 Conclusion

Although heuristics are frequently tailored for a given problem domain, and in some cases even for a given instance, the high level hyper-heuristic methodologies are more general and their components are reusable without requiring any algorithmic modification while dealing with unseen instances. There is a growing number of studies on such search algorithms. A selection type of hyper-heuristic commonly manages the search process by controlling a set of low level heuristics or move operators and their parameters. In this study, we present a stochastic search method which is significantly enhanced by a selection hyper-heuristic under a generalised iterated local search framework. The multistage approach switches between diversification and intensification processes automatically and allows partial restarts via a threshold move acceptance method whose parameter is also

Table 4 The best-of-runs performance comparison of the HySST approach to the other competing approaches using the quality (cost) of a solution indicated as feasibility-value.objective-value in Round 3. The best values are highlighted in bold.

Dataset	HySST	GOAL	HFT	Lectio	VAGOS
Instance2	0.00044	0.00032	0.00082	**0.00005**	0.00026
Instance3	0.00084	0.00101	0.00212	0.00048	**0.00047**
Instance4 - Brazil	0.00176	1.00136	0.00205	0.00090	**0.00078**
Instance6	0.00150	0.00160	0.00347	**0.00060**	0.00074
ElementarySchool	**0.00003**	**0.00003**	**0.00003**	**0.00003**	ABSENT
SecondarySchool2	**0.00000**	**0.00000**	0.00576	**0.00000**	ABSENT
Aigio 1st HS 2010	0.00218	**0.00000**	0.00555	0.00076	ABSENT
Instance4 - Italy	**0.00052**	0.00061	0.08623	0.00078	ABSENT
Instance1	0.01721	**0.00003**	36.12987	274.00281	ABSENT
Kottenpark2003	0.03919	0.05355	1.88983	**0.02918**	ABSENT
Kottenpark2005A	**15.28693**	24.13930	36.36132	198.04845	ABSENT
Kottenpark2008	16.17720	**10.27909**	167.99999	129.69216	ABSENT
Kottenpark2009	**18.08010**	19.05590	148.99999	87.09440	ABSENT
Woodlands2009	0.00013	**0.00012**	8.00206	0.00019	ABSENT
School	0.00920	**0.00441**	1.08163	0.00762	ABSENT
WesternGreeceUni3	0.00007	**0.00005**	0.00032	30.00002	**0.00005**
WesternGreeceUni4	0.00009	**0.00008**	0.00142	35.00058	ABSENT
WesternGreeceUni5	**0.00000**	**0.00000**	0.00064	4.00001	**0.00000**
Average ranking	2.25	**1.64**	3.75	2.75	3.86

controlled by the proposed method. We joined the ITC2011 competition, as the team "HySST" (Hyper-heuristic Search Strategies and Timetabling), with this multistage selection hyper-heuristic.

A selection hyper-heuristic should be "fast to implement, requiring far less expertise in either the problem domain or heuristic methods, and robust enough to effectively handle a range of problems" (Cowling et al, 2001). The teams HFT, Lectio and VAGOS attempted to develop tailored solutions in the given the limited time, while the HySST team preferred implementing a hyper-heuristic. Ultimately, our approach performed better than the approaches proposed by those teams, though couldn't beat the approach proposed by GOAL. However, the primary point of our work is that it shows the utility of the hyper-heuristic in that it makes better usage of the domain specific heuristics, and in particular demonstrates the advantages of multistage methods with adaptive relaxations.

Acknowledgement

This work is supported in part by the UK EPSRC under grant EP/F033214/1 - The LANCS Initiative in Foundational Operational Research: Building Theory for Practice.

References

Abramson D (1991) Constructing school timetables using simulated annealing: Sequential and parallel algorithms. Management Science 37(1):98–113

Abramson D, Dang H (1993) School timetables: A case study in simulated annealing. In: Applied Simulated Annealing, Lecture Notes in Economics and Mathematical Systems, vol 396, Springer Berlin Heidelberg, pp 103–124

Abramson D, Dang H, Krisnamoorthy M (1999) Simulated annealing cooling schedules for the school timetabling problem. Asia-Pacific Journal of Operational Research 16:1–22

Beligiannis GN, Moschopoulos CN, Kaperonis GP, Likothanassis SD (2008) Applying evolutionary computation to the school timetabling problem: The greek case. Computers & Operations Research 35(4):1265 – 1280

Bello GS, Rangel MC, Boeres MCS (2008) An Approach for the Class /Teacher Timetabling Problem. In: PATAT '08 Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling

Birbas T, Daskalaki S, Housos E (2009) School timetabling for quality student and teacher schedules. J of Scheduling 12(2):177–197

Broder S (1964) Final examination scheduling. Commun ACM 7(8):494–498

Burke EK, Landa-Silva JD, Soubeiga E (2005) Meta-heuristics: Progress as Real Problem Solvers, Springer, chap Multi-objective Hyper-heuristic Approaches for Space Allocation and Timetabling, pp 129–158,

Burke EK, Curtois T, Hyde MR, Kendall G, Ochoa G, Petrovic S, Rodríguez JAV, Gendreau M (2010a) Iterated local search vs. hyper-heuristics: Towards general-purpose search algorithms. In: IEEE Congress on Evolutionary Computation, pp 1–8

Burke EK, Hyde M, Kendall G, Ochoa G, Özcan E, Woodward JR (2010b) A classification of hyper-heuristics approaches. In: Gendreau M, Potvin JY (eds) Handbook of Metaheuristics, International Series in Operations Research & Management Science, vol 57, 2nd edn, Springer, chap 15, pp 449–468

Burke EK, Gendreau M, Hyde M, Kendall G, Ochoa G, Özcan E, Qu R (2013) Hyper-heuristics: a survey of the state of the art. Journal of the Operational Research Society 64(12):1695–1724

Chakhlevitch K, Cowling P (2005) Choosing the fittest subset of low level heuristics in a hyperheuristic framework. In: Proceedings of 5th European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP2005), Springer, Lecture Notes in Computer Science, vol 3448, pp 25–33

Chakhlevitch K, Cowling P (2008) Hyperheuristics: Recent developments. In: Cotta C, Sevaux M, Srensen K (eds) Adaptive and Multilevel Metaheuristics, Studies in Computational Intelligence, vol 136, Springer Berlin Heidelberg, pp 3–29

Colorni A, Dorigo M, Maniezzo V (1992) A genetic algorithm to solve the timetable problem. Tech. Rep. Technical Report No. 90-060, Politecnico di Milano, Italy

Corne D, Ross P, Fang HL (1994) Fast practical evolutionary timetabling. In: Selected Papers from AISB Workshop on Evolutionary Computing, Springer-Verlag, London, UK, UK, pp 250–263

Cowling P, Chakhlevitch K (2003) Hyperheuristics for managing a large collection of low level heuristics to schedule personnel. In: Proceedings of the 2003 Congress on Evolutionary Computation, pp 1214–1221

Cowling P, Kendall G, Soubeiga E (2001) A hyperheuristic approach to scheduling a sales summit. In: Selected papers from the Third International Conference on Practice and Theory of Automated Timetabling, Springer-Verlag, London, UK, pp 176–190

Erben W, Keppler J (1996) A genetic algorithm solving a weekly course-timetabling problem. In: Burke E, Ross P (eds) Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science, vol 1153, Springer Berlin Heidelberg, pp 198–211

Even S, Itai A, Shamir A (1976) On the Complexity of Timetable and Multicommodity Flow Problems. SIAM Journal on Computing 5(4):691–703

Filho GR, Antonio L, Lorena LAN (2001) A constructive evolutionary approach to school timetabling. In: In Applications of Evolutionary Computing, Springer Lecture, pp 130–139

Fisher H, Thompson GL (1963) Probabilistic learning combinations of local job-shop scheduling rules. In: Muth JF, Thompson GL (eds) Industrial Scheduling, Prentice-Hall, Inc, New Jersey, pp 225–251

Hertz A (1992) Finding a feasible course schedule using a tabu search. Discrete Applied Mathematics 35:255–270

Jacobsen F, Bortfeldt A, Gehring H (2006) Timetabling at German Secondary Schools: Tabu Search versus Constraint Programming. In: in Proceedings 6th international conference on the practice and theory of automated timetabling, PATAT2006, pp 439–442

Kannan A, van den Berg G, Kuo A (2012) ischedule to personalize learning. Interfaces 42(5):437–448

Kautz H, Horvitz E, Ruan Y, Gomes C, Selman B (2002) Dynamic restart policies. In: Eighteenth national conference on Artificial intelligence, American Association for Artificial Intelligence, Menlo Park, CA, USA, pp 674–681

Kendall G, Mohamad M (2004) Channel assignment optimisation using a hyper-heuristic. In: Proceedings of the 2004 IEEE Conference on Cybernetic and Intelligent Systems (CIS2004), Singapore, pp 790–795

Kingston J (2005) A tiling algorithm for high school timetabling. In: Burke E, Trick M (eds) Practice and Theory of Automated Timetabling V, Lecture Notes in Computer Science, vol 3616, Springer Berlin Heidelberg, pp 208–225

Lourenço HR, Martin OC, Stützle T (2010) Iterated local search: Framework and applications. In: Gendreau M, Potvin JY (eds) Handbook of Metaheuristics, International Series in Operations Research and Management Science, vol 146, Springer US, pp 363–397

Luby M, Sinclair A, Zuckerman D (1993) Optimal speedup of Las Vegas algorithms. Information Processing Letters 47(4):173–180

Marte M (2007) Towards constraint-based school timetabling. Annals of Operations Research 155:207–225

McCollum B, Schaerf A, Paechter B, McMullan P, Lewis R, Parkes AJ, Gaspero LD, Qu R, Burke EK (2010) Setting the research agenda in automated timetabling: The second international timetabling competition. INFORMS J on Computing 22(1):120–130

Moura AV, Scaraficci RA (2010) A GRASP strategy for a more constrained School Timetabling Problem. International Journal of Operational Research 7

Mustafa M (2012) Intelligent hyperheuristics: A tool for solving generic optimisation problems. PhD thesis, Department of Computer Science, KU Leuven, Belgium

Özcan E, Kheiri A (2011) A hyper-heuristic based on random gradient, greedy and dominance. In: Gelenbe E, Lent R, Sakellari G (eds) ISCIS, Springer, pp 557–563

Özcan E, Bilgin B, Korkmaz EE (2008) A comprehensive analysis of hyper-heuristics. Intelligent Data Analysis 12(1):3–23
Özcan E, Bykov Y, Birben M, Burke E (2009) Examination timetabling using late acceptance hyper-heuristics. In: Evolutionary Computation, 2009. CEC '09. IEEE Congress on, pp 997 –1004
Özcan E, Parkes AJ, Alkan A (2012) The interleaved constructive memetic algorithm and its application to timetabling. Comput Oper Res 39(10):2310–2322
Pillay N (2010a) An overview of school timetabling research. In: Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2010), pp 321–335
Pillay N (2010b) A study into the use of hyper-heuristics to solve the school timetabling problem. In: Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists, ACM, New York, NY, USA, SAICSIT '10, pp 258–264
Pillay N (2012) Hyper-heuristics for educational timetabling. In: Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)
Pisinger D, Ropke S (2007) A general heuristic for vehicle routing problems. Computers and Operations Research 34:2403– 2435
Post G, Ahmadi S, Daskalaki S, Kingston JH, 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, Gaspero LD, Kingston JH, McCollum B, Schaerf A (2012) The third international timetabling competition. In: Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)
Raghavjee R, Pillay N (2008) An application of genetic algorithms to the school timetabling problem. In: Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology, ACM, New York, NY, USA, SAICSIT '08, pp 193–199
Raghavjee R, Pillay N (2010) Using genetic algorithms to solve the South African school timetabling problem. In: Second World Congress on Nature & Biologically Inspired Computing, NaBIC 2010, 15-17 December 2010, Kitakyushu, Japan, IEEE, pp 286–292
Ross P (2005) Hyper-heuristics. In: Burke EK, Kendall G (eds) Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, Springer, chap 17, pp 529–556
Ross P, Corne D, Fang HL (1994) Improving evolutionary timetabling with delta evaluation and directed mutation. In: Davidor Y, Schwefel HP, Männer R (eds) Parallel Problem Solving from Nature, PPSN III, Lecture Notes in Computer Science, vol 866, Springer Berlin / Heidelberg, pp 556–565
Schaerf A (1996) Tabu search techniques for large high-school timetabling problems. In: Proceedings of the thirteenth national conference on Artificial intelligence - Volume 1, AAAI Press, AAAI'96, pp 363–368
Smith KA, Abramson D, Duke D (2003) Hopfield neural networks for timetabling: formulations, methods, and comparative results. Comput Ind Eng 44(2):283–305
Valouxis C, Housos E (2003) Constraint programming approach for school timetabling. Computers and Operations Research 30(10):1555–1572

de Werra D (1997) The combinatorics of timetabling. European Journal of Operational Research 96(3):504–513

Wilke P, Gröbner M, Oster N (2002) A hybrid genetic algorithm for school timetabling. In: Proceedings of the 15th Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence, Springer-Verlag, London, UK, UK, AI '02, pp 455–464