# An Online Learning Selection Hyper-heuristic for Educational Timetabling

**Alexander Steenson · Ender Özcan · Ahmed Kheiri · Barry McCollum · Paul McMullan**

**Abstract** Examination and course timetabling are computationally difficult real-world resource allocation problems. In 2007, an International Timetabling Competition (ITC) consisting of three classes; (i) examination timetabling, (ii) post enrollment-based, and (iii) curriculum-based course timetabling was organised. One of the competing algorithms, referred to as CPSolver, successfully achieved the first place in two out of these three tracks. This study investigates the performance of various multi-stage selection hyper-heuristics sequencing low-level heuristics/operators extending the CPSolver framework which executes hill climbing and two well-known local search metaheuristics in stages. The proposed selection hyper-heuristic is a multi-stage approach making use of a matrix which maintains transitional probabilities between each low-level heuristic to select the next heuristic in the sequence. A second matrix tracks the probabilities of ending the sequence on a given low-level heuristic. The best configuration for the selection hyper-heuristic is explored tailoring the heuristic selection process for the given timetabling problem class. The empirical results on the ITC 2007 problem instances show that the proposed selection hyper-heuristics can reduce the number of soft constraint violations, producing improved solutions over CPSolver as well as some other previously proposed solvers, particularly, in examination and curriculum-based course timetabling.

A. Steenson (corresponding author), E. Özcan
University of Nottingham, Nottingham, UK
E-mail: {psyajs,ender.ozcan}@nottingham.ac.uk

A. Kheiri
Lancaster University, Lancaster, UK
E-mail: a.kheiri@lancaster.ac.uk

B. McCollum, P. McMullan
Queen's University, Belfast, UK
E-mail: {b.mccollum,p.mccmullan}@qub.ac.uk

## 1 Introduction

Educational timetabling represents a class of challenging real-world problems which are still of interest to many researchers and practitioners. The International Timetabling Competition (ITC) 2007 provided researchers with models of the problems faced, which incorporate an increased number of real-world constraints[1]. In 2007 the ITC was composed of three tracks[2]:

- **Examination Timetabling** where a set of exams must be assigned into a predefined examination time period.
- **Post Enrolment-based Course Timetabling** where a course timetable must be constructed after students have selected and enrolled onto a set of courses.
- **Curriculum-based Course Timetabling** where university lectures, for several courses, must be assigned a room and timeslot within a set of restrictions.

The tracks describe common problems that universities face around the world when scheduling exams or courses. When scheduling these timetables, no hard constraints and as little soft constraints as possible should be violated. Violating no hard constraints is the top priority when creating a timetable, in order to make it feasible and therefore workable. For example, whilst scheduling for an examination timetable, no student can be assigned to take more than one exam at any given time, and no more students than available seats can be assigned to a given room. Soft constraints however, may be violated and still result in a feasible timetable. Soft constraints represent preferences whilst scheduling, to aid in the smooth running and efficiency of the timetable. An example of a soft constraint in the Examination track is ‘two exams in a row’; it is preferred if students would not take multiple exams ‘back to back’. Therefore, the aim is to minimise soft constraints whilst not violating hard constraints.

The ITC 2007 released a number of data instances to the research community throughout the duration of the competition. This gave researchers the opportunity to develop new and innovative approaches to solving the problems outlined in each track, on real life datasets. As a result, many successful methods have been developed to solve these types of problems. Five finalists were selected based on their submitted results on the released datasets. The solvers developed by the finalists were then judged further using a set of ‘hidden’ datasets that were not released for testing and development, and again using the previously released datasets. It is important to note that computation time was limited on each solver ensuring a fair outcome. The winner of the Examination track and the Curriculum-based Timetabling track was Thomàš Müller, who developed a hybrid constraint-based solver, known as CPSolver, as part of his PhD to compete in all three competition tracks [23]. This single solution framework was capable of constructing and refining solutions for each

[1] http://www.cs.qub.ac.uk/itc2007/index.htm
[2] http://www.cs.qub.ac.uk/itc2007/index_files/competitiontracks.htm

track, by employing a series of algorithms, including hill climbing and local search metaheuristics [29], namely, Great Deluge and Simulated Annealing, in stages, where each operate over feasible, though not necessarily complete solutions. Those algorithmic components may need to be heavily modified and reconfigured to be applied to another problem [17].

Hyper-heuristics [10] are high-level control methods used in heuristic optimisation that operate over the search space formed by a set of low-level heuristics rather than the search space of solutions, directly. The hyper-heuristics are classified into two as either being generative (generating low-level heuristics) or selective (selecting from a pool of low-level heuristics). This paper will be focusing purely on the latter one. A selection hyper-heuristic attempts to choose the right method or sequence of (meta)heuristics in a given situation at each step or stage during the search process [4,12]. This paper investigates the effects of implementing a sequence-based heuristic selection algorithm within Thomàš Müller's CPSolver framework to uncover Hidden Markov chains and generate heuristic sequences tailored to the problem instance. This paper provides an explanation of the implementation of the sequence-based hyper-heuristic within the CPSolver framework. Finally, results regarding the effectiveness of the implemented hyper-heuristic against Thomàš Müllers original ITC 2007 results, along with the results achieved by the other ITC 2007 finalists and some post ITC 2007 solutions can be found at the end of this paper.

## 2 Related Work

Many approaches have been taken to solve the problem instances provided as part of the competition. These approaches range from hyper-heuristics capable of solving each instance from each track with minimal domain specific requirements, to domain-specific solvers which are only capable of solving the instances of one given track. A number of different methods have been developed to solve a problem for each track individually. For the Examination track the following approaches have been proposed: graph colouring constructive hyper-heuristic [28], tabu search [11], simulated annealing [13] and cell division [27]. For the Post Enrolment-based Course Timetabling track: local search [6,8,30,31], ant colony [25], simulated annealing [7,20] and tabu search [14]. Finally for the Curriculum-based Course Timetabling: adaptive tabu search [22], iterative local search [22], threshold accepting metaheuristic [16] and repair-based heuristic search [9].

As mentioned above, one hyper-heuristic approach, winning two tracks was Thomàš Müller's three phase constraint-based solver [23]. This single solution framework was capable of constructing and refining solutions for each track, by employing a series of algorithms based on local search techniques that operate over feasible, though not necessarily complete, solutions. The framework only ever operates over a feasible solution space, ensuring all hard constraints are satisfied, by using a series of algorithms operating using local search techniques. The CPSolver framework consists of multiple phases run sequentially, namely the construction phase, the Hill Climbing (HC) phase, the Great Deluge (GD) phase and finally the Simulated Annealing (SA) phase. The construction phase generates a complete initial solution by performing an iterative forward search algorithm that uses conflict-based statistics [23]. A local optimum is then obtained using hill climbing and once an improving solution can no longer be found the great deluge phase begins. Oscillations of the bounds within great deluge are used to allow worsening solutions, facilitating the escape of the local optimum. The simulated annealing stage is optional and is not used within the Examination track to increase the speed of convergence. Once simulated annealing has met its termination criteria the hill climbing phase is continued to converge back to a local optimum. As a consequence, the framework constructs a feasible solution and proceeds to improve upon it.

Implemented within each of the perturbative phases are a number of low-level neighbourhood heuristics selected at random. These low-level heuristics are one of the few domain-specific requirements needed within the framework. This paper seeks to improve on this section of the solver by removing the random aspect of selecting low-level heuristics by introducing online learning to create sequences of low-level heuristics to be applied to the current solution. By removing the random selection, it is possible to target low-level heuristics that perform better than others, decreasing the number of iterations needed to achieve a good solution value. This technique has not before been applied to solve the ITC 2007 problem instances over all thee tracks.

Whilst CPSolver method won the Examination track and Curriculum-based Timetabling track, it came fifth in the Post Enrolment-based Course Timetabling track. It is therefore worth noting the techniques capable of producing higher quality solutions.

One notable technique put forward by Atsuta [1] applied a general-purpose solver. Problem data instances were represented as linear 0-1 inequalities, quadratic 0-1 inequalities, and all-different constraints. Using predetermined weights for hard and soft constraints, a tabu search combined with an iterated local search is used to solve the given instance. The constraint weightings are dynamically controlled during the search process to improve the performance of the general-purpose CPS. It is also worth noting that this technique is not domain specific, and to demonstrate the capabilities of the solver it was entered into each track of the competition producing high quality results on all three tracks. The solver was placed third in the Examination and Curriculum-based tracks, and achieved second place in the Post Enrolment track.

Since the end of the second International Timetabling Competition there have been a number of papers released outlining techniques used to solve the problem instances used within the competition. One of the advantages for researchers evaluating techniques post competition is that all the datasets are available, allowing researchers to tailor frameworks to the data without the time pressure of the competition. On six of the Examination datasets, a technique used to provide lower solution values than all other submissions was proposed in [5]. Here a two-stage solver was implemented. The first stage of the solver employs a construction algorithm using the existing adaptive ordering heuristic [3]. The next stage aims to improve the solution created in the first stage. An extended great deluge algorithm technique is used, applying a reheating mechanism after a set number of non-improving moves. They performed 51 runs for each data instance within the ITC 2007 Examination track to compare with the other competitors' 10 runs of each data instance in the ITC 2007. However, they do acknowledge this in the paper. It is also not clear on the specifications of the computer the instances were solved on, and there is no mention if the benchmarking program issued by the ITC 2007 was used to ensure a fair comparison.

## 3 A Multi-stage Sequence-based Hyper-heuristic Approach

Müller [23]'s approach to timetabling is a multi-stage approach using the CPSolver framework, where a different search algorithm is utilised at each fixed stage. Also, this approach used a different set of low-level heuristics/operators implemented for each timetabling problem class in the ITC 2007 competition. At each step in a stage, a random heuristic is selected and applied to the incumbent solution producing a new solution. To maximise the possibility of making even further improvements to the newly created solutions, learning can be utilised guiding the heuristic selection. Hence the random heuristic selection method is altered to incorporate online learning. Online learning techniques allow for learning to take place whilst a problem instance is being solved. Given that there are three tracks presented within the competition, each will have multiple problem instances with different characteristics, hence online learning is likely to improve the performance of a no-learning approach Incorporating an online learning method into the selection hyper-heuristic ordering the execution of low-level heuristics can generate complex sequences (which would correspond to new heuristics/operators) enabling creation of improved solutions in reduced time.

The proposed approach is based on the same CPSolver, hence it is still a multi-stage approach. The selection hyper-heuristics employed at each stage is not changed and assigns a score to each LLH, maintains those scores based on reinforcement learning and then it chooses one of the low-level heuristics based on their scores using the heuristic selection method. The components and variants of the proposed approach is described in the following subsections.

### 3.1 Low-level Heuristics

The same set of low-level heuristics (LLHs) as suggested in [23] are employed for solving the timetabling problems as summarised below for each ITC 2007 track.

1. Examination timetabling: Exam swap, period swap, room swap, period change, room change, period and room change.

2. Post enrollment-based course timetabling: Time move, room move, event move, event swap, precedence swap (violation oriented move with considering the precedence constraints).
3. Curriculum-based course timetabling: Time move, room move, lecture move, room stability move, min working days move, curriculum compactness move.

## 3.2 Heuristic Selection

Kheiri and Keedwell [17] provided an effective sequence-based selection hyper-heuristic illustrated its performance across a number of high school timetabling problem instances.

Commonly, sequence-based heuristic selection algorithms work by generating a completed sequence of low-level heuristic (LLH) operators to be sequentially applied to the current solution, creating a new candidate solution. This can lead to the scenario of constructing a sequence of length $n$, applying each LLH to the current solution and reverting back if the solution value is not accepted by the acceptance strategy, e.g. simulated annealing. In an attempt to eliminate reverting back $n$ neighbourhood changes when a sequence is rejected, when the next LLH in the sequence is selected it is immediately applied to find a neighbouring solution. If the neighbouring solution is accepted, it is officially added to the sequence, otherwise the LLH is not added to the current sequence, and the next LLH is selected. Implementing the sequence construction in this manner could help guide the sequence creation process and discovery of Hidden Markov Chains.

Sequences are constructed by maintaining two matrices. The first matrix is designed to maintain a performance score for each possible heuristic transition, we will refer to this as $TransScore$. We are then capable of selecting the next LLH using these scores. For example, assume $n$ low-level heuristics are implemented. Also, assume the unfinished sequence of $[llh_i, llh_j]$ is constructed and the next LLH selected, using a Roulette Wheel or any other heuristic selector, is $llh_k$. From this, assume heuristic $llh_k$ is applied and produced an improved solution. The score value at $TransScore[j, k]$ is updated and the current sequence becomes $[llh_i, llh_j, llh_k]$. If $llh_k$ did not produce an improving solution but was still accepted, the score values remain the same and the sequence is again updated to become $[llh_i, llh_j, llh_k]$. The starting scores for each heuristic transition is 1, that is, $TransScore[llh_i, llh_j] = 1, \forall\ i, j$.

The second matrix stores the score values of ending the sequence on the current LLH selected. We will refer to this matrix as $EndScore$. For example, with the same implementation and sequence construction in the example above, when $llh_k$ is selected the scores at $EndScore[llh_k, 0]$ and $EndScore[llh_k, 1]$ are used to determine if the sequence is terminated at that heuristic. If the sequence is to terminate at $llh_k$ then the scores $EndScore[i, 0]$, $EndScore[j, 0]$ and $EndScore[k, 1]$ are all updated. The starting scores for each acceptance strategy for every low-level heuristic is 1, that is, $EndScore[llh_i, j] = 1, \forall\ i, j$.

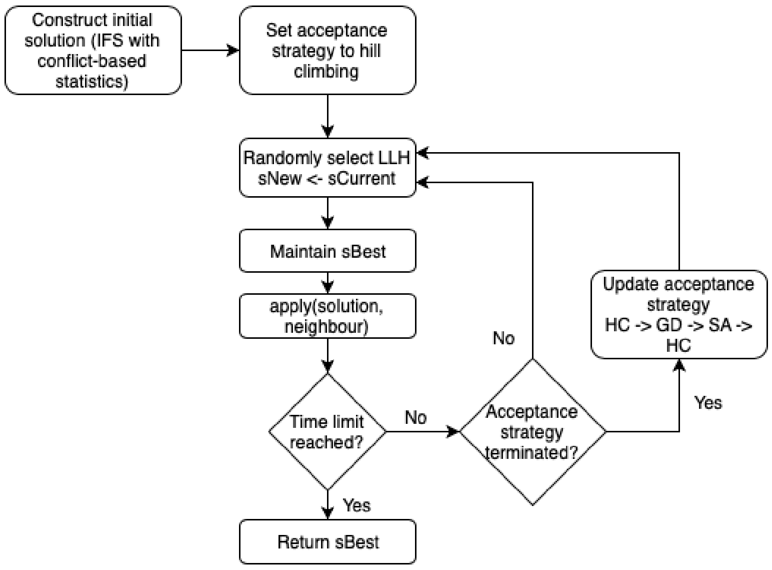
Construct initial solution (IFS with conflict-based statistics)
Set acceptance strategy to hill climbing
Randomly select LLH sNew <- sCurrent
Maintain sBest
apply(solution, neighbour)
Time limit reached?
No
Yes
Return sBest
Acceptance strategy terminated?
No
Yes
Update acceptance strategy HC -> GD -> SA -> HC

Fig. 1: Flowchart of the CPSolver

This learning method is implemented into the CPSolver framework to select heuristics instead of the current random selection. The figures displayed below compare the stages that the CPSolver takes in the 2007 ITC submission against the stages the CPSolver takes with the new proposed implementation.

Given the two scoring probability matrices described above, a selection strategy is required to select the next LLH in the sequence and to determine if the sequence is terminated. Generally, the selection process can be categorised into proportional and elitist strategies. Proportional strategies take into consideration the probabilities of each heuristic being selected and selects the heuristic accordingly. This allows diversification within the heuristics selected but takes longer to converge. Elitist methods simply take the heuristic with the best score associated to it. This increases convergence but eliminates any diversification throughout the search. The heuristic selection strategies explored in this paper are: Roulette Wheel selection and Tournament selection. Whilst a general aim is to reduce computation time of the solver, and both of these selection methods fall under the proportional strategy, taking the heuristic with the highest score each time does not allow for enough diversification for the problem instances.

Roulette Wheel selection (RW) gives an individual $i$ the probability of being selected $p(i)$ proportional to its fitness $f(i)$ where [15]:

$$p(i) = \frac{f(i)}{\sum_{j=1}^{n} f(j)} \tag{1}$$

Tournament selection (TO) consists of selecting $k$ individuals at random and then ranking them best to worst. The best individual is then selected to

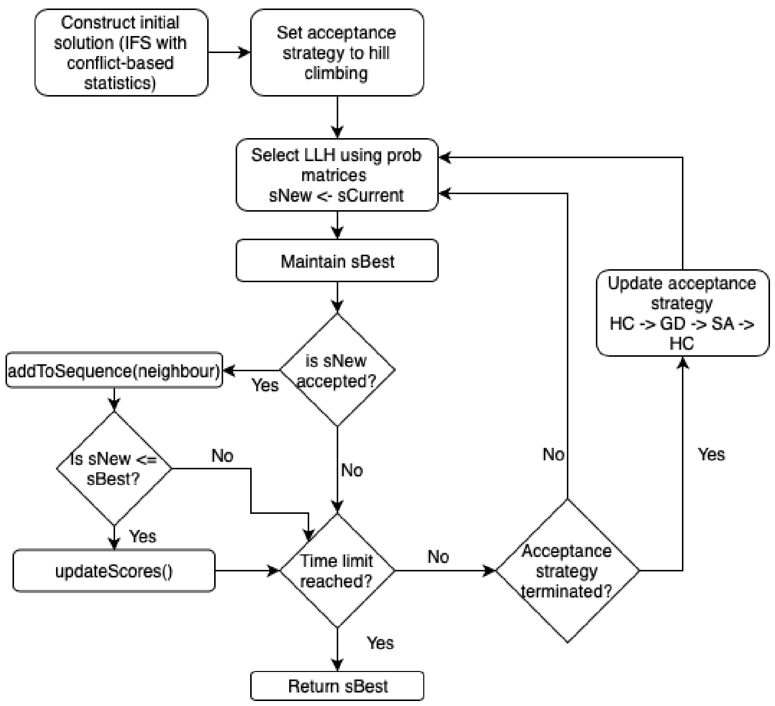
Construct initial solution (IFS with conflict-based statistics)
Set acceptance strategy to hill climbing
Select LLH using prob matrices sNew <- sCurrent
Maintain sBest
Update acceptance strategy HC -> GD -> SA -> HC
is sNew accepted?
addToSequence(neighbour)
Yes
Is sNew <= sBest?
No
No
No
Yes
Yes
updateScores()
Time limit reached?
No
Acceptance strategy terminated?
Yes
Return sBest

Fig. 2: Flowchart of the new sequence-based hyper-heuristic solver

enter the sequence [15]. Tournament selection has the advantage of maintaining diversification whilst converging faster than Roulette Wheel.

## 3.3 Reward Scheme

The performance of the sequence-based hyper-heuristic can vary depending on the learning strategy implemented. This paper explores three rewarding schemes as explained below [17].

- Linear reward scheme (LI): The matrices values are incremented linearly with a reward of 1 when an improving solution is obtained. The strategy does not take into consideration the amount to which the solution is improved, but simply acknowledges that an improving solution has been found.
- Non-linear reward scheme (NL): The matrices values are incremented non-linearly by $e^t/c$, where $t$ is the time elapsed and $c$ is a predetermined constant. This reward system allocates a larger reward to sequences that find an improving move later on in the search.
- Delta reward scheme (DE): The matrices values are incremented according to how much the sequence improved on the overall solution value, giving

larger rewards to sequences that make a bigger impact on the solution value. The heuristic selection can then continue making sequences that target the heuristics that make the biggest difference to the solution value.

## 3.4 Stage Options

CPSolver is used as a multi-stage approach, constructing a (feasible) solution and then improving upon it using a hill climbing and/or a local search meta-heuristic. During the construction phase, the solver can sometimes run into the issue where it can no longer successfully assign values to variables causing it to be idle. This is very noticeable on data instances 3 and 4 on the Examination track, with Müller being unable to obtain a feasible solution on some of his runs. To combat this issue, if the construction phase does not make an improving move in 200 iterations the current solution will be reset. The construction phase will then restart and build a new solution. This method proved to be reliable as we managed to obtain feasible solutions on all data instances in the Examination track.

After the construction phase, the hyper-heuristic follows a fixed stage structure of Hill Climbing (HC), Great Deluge with re-rising level (GD) and then the optional stage Simulated Annealing with reheating (SA) (Thomàš Müller noted that Simulated Annealing is not used for the Examination timetabling track due to its slow computation time [23]). A selection hyper-heuristic contains two key components: heuristic selection and move acceptance. CPSolver provides a multi-stage selection hyper-heuristic framework where a different move acceptance method can be used at each stage while the heuristic selection is maintained. The move acceptance methods significantly affects the performance as compared to heuristic selection within hyper-heuristics [26]. Since we have only 2 move acceptance methods, we explored the performance of the proposed approach with different options for the stages where we either use HC or not or a move acceptance. Hence every stage ordering using (a set of) two or single move acceptance method(s) is tested along with using HC or not in the first stage, yielding six different options: HC-GA-SA, HC-SA-GD, GD-SA, SA-GD, GD and finally SA.

The same suggested parameter settings for each ITC2007 track provided in [23] for the HC (maximum number of iterations without improvement), GD (upper bound for the level, lower bound for the level, rate of decrease), SA (initial temperature, number of steps spent at each temperature, geometric cooling rate, reheating rate) algorithms are used during the experiments.

## 4 Configuring the Sequence-based Hyper-heuristic Approach

All configurations of the sequence-based hyper-heuristic approach are evaluated using 4 selected problem instances from each track of the competition. Each experiment is run 10 times for the appropriate amount of time (247 seconds) which was obtained using the benchmarking software supplied by the

ITC 2007. This helps create a level playing field across all competition submissions and allows for the comparison of results without having to factor in computer speed. A total of 132 hours were spent configuring our hyper-heuristic approach, and all the runs were performed on a Windows 10 machine with an Intel Core i5-6400 2.70GHz processor with 8GB of RAM. When testing for the acceptance strategy, heuristic selection strategy and reward mechanism, the data instances used for the Examination timetabling track were 1, 2, 5 and 6. For the Post Enrolment timetabling track the data instances 3, 4, 6 and 8 were used. Finally, for the Curriculum-based timetabling track the data instances 3, 5, 7 and 8 were used. To score the results of each run, a given instance was allocated a rank based on its overall solution value. The solution value with the lowest score was given a ranking score of 1, the next highest will get a ranking score of 2 etc. In the case of a tie an average rank was assigned. For example, if the third highest solution value was 5 and this solution value was obtained 3 times, a ranking score of 4 would be given to each solution. The ranking scores would then continue from 6. The score for a given strategy on a given instance is the mean ranking score for that instance. The overall score for a given strategy is then given by the mean score obtained for each data instance, and the strategy with the lowest score is then taken to be the best.

### 4.1 Stage Option Experiments

The incremental configuration for the proposed approach first experimented with different stage options: HC-GA-SA, HC-SA-GD, GD-SA, SA-GD, GD and finally SA. For all three tracks the heuristic selection method was set to Roulette Wheel with a linear reward method for each option. Table 1 shows the ranking of each stage option for each selected benchmark instance for each track. Table 2 shows the overall ranking of the acceptance strategies for each track. From the tables, we can see for all three tracks, Great Deluge by itself performed the best. However, Table 2 demonstrates that the second best stage option is HC-GD-SA, GD-SA and SA-GD for the examination, post enrollment-based and curriculum-based course timetabling, respectively. Hence, those methods are fixed for the next set of experiments.

### 4.2 Testing the Heuristic Selection Methods and Reward Mechanisms

During implementation it was noticed that the heuristic selection method and reward mechanism were closely related with the combination dramatically affecting the scores. This means we were not able to optimise them sequentially, they had to be done simultaneously. We also tested each combination of heuristic selection and reward mechanism for the top two ranking sequence of acceptance strategies obtained from the section above. Table 3 shows the overall ranked scores for each track. We can see from the table that the Examination track performed best by a significant margin, operating with a delta-based

Table 1: Acceptance strategy ordering per instance ranking

| Examination Acceptance Strategy Ordering | | | | | | | |
|---|---|---|---|---|---|---|---|
| Instance 1 | | Instance 2 | | Instance 5 | | Instance 6 | |
| Order | Score | Order | Score | Order | Score | Order | Score |
| GD | 9.30 | SA | 21.60 | GD | 5.60 | GD | 22.20 |
| HC-GD-SA | 12.70 | GD | 23.60 | SA | 23.00 | GD-SA | 28.35 |
| SA | 35.75 | HC-GD-SA | 27.15 | HC-GD-SA | 30.15 | SA | 30.90 |
| SA-GD | 37.10 | HC-SA-GD | 31.40 | GD-SA | 40.05 | HC-GD-SA | 31.40 |
| GD-SA | 40.15 | SA-GD | 37.30 | SA-GD | 40.60 | SA-GD | 34.65 |
| HC-SA-GD | 48.00 | GD-SA | 41.95 | HC-SA-GD | 43.60 | HC-SA-GD | 35.50 |
| Post Enrolment Acceptance Strategy Ordering | | | | | | | |
| Instance 3 | | Instance 4 | | Instance 6 | | Instance 8 | |
| Order | Score | Order | Score | Order | Score | Order | Score |
| GD | 17.20 | GD | 18.05 | GD | 17.40 | GD | 12.75 |
| HC-GD-SA | 18.45 | HC-GD-SA | 25.00 | GD-SA | 22.90 | GD-SA | 28.00 |
| GD-SA | 21.25 | GD-SA | 26.95 | HC-GD-SA | 27.30 | HC-SA-GD | 28.65 |
| SA-GD | 35.20 | SA-GD | 27.40 | HC-SA-GD | 30.05 | SA-GD | 29.30 |
| HC-SA-GD | 36.00 | HC-SA-GD | 33.40 | SA-GD | 36.00 | HC-GD-SA | 29.70 |
| SA | 54.90 | SA | 52.20 | SA | 49.35 | SA | 54.60 |
| Curriculum-based Acceptance Strategy Ordering | | | | | | | |
| Instance 3 | | Instance 5 | | Instance 7 | | Instance 8 | |
| Order | Score | Order | Score | Order | Score | Order | Score |
| GD | 18.60 | GD | 17.35 | HC-SA-GD | 19.65 | GD | 16.65 |
| GD-SA | 22.30 | GD-SA | 22.40 | GD | 19.70 | SA-GD | 16.65 |
| SA-GD | 29.55 | SA-GD | 28.70 | SA-GD | 22.90 | HC-SA-GD | 30.45 |
| HC-SA-GD | 32.60 | HC-GD-SA | 31.30 | SA | 33.30 | GD-SA | 38.35 |
| HC-GD-SA | 34.70 | HC-SA-GD | 33.05 | GD-SA | 43.65 | SA | 39.30 |
| SA | 45.25 | SA | 50.20 | HC-GD-SA | 43.80 | HC-GD-SA | 41.60 |

Table 2: Acceptance strategy ordering overall ranking

| Examination | | Post Enrolment | | Curriculum-based | |
|---|---|---|---|---|---|
| Order | Score | Order | Score | Order | Score |
| GD | 15.18 | GD | 16.35 | GD | 18.07 |
| HC-GD-SA | 25.35 | GD-SA | 24.77 | SA-GD | 24.45 |
| SA | 27.81 | HC-GD-SA | 25.11 | HC-SA-GD | 28.94 |
| SA-GD | 37.41 | SA-GD | 31.98 | GD-SA | 31.68 |
| GD-SA | 37.62 | HC-SA-GD | 32.02 | HC-GD-SA | 37.85 |
| HC-SA-GD | 39.62 | SA | 52.76 | SA | 42.01 |

reward system with tournament heuristic selection and with Great Deluge as the move acceptance. The Post Enrolment track performed best with a delta-based reward system with Tournament heuristic selection and with Great Deluge followed by Simulated Annealing as the move acceptance ordering. It is interesting to note that Great Deluge followed by Simulated Annealing ranked second best during the move acceptance optimisation stage, running under a linear reward with Roulette Wheel selection. This further indicates that the performance reward mechanism and heuristic selection can be dependent on the move acceptance. Optimising for one feature at a time may not lead to the optimal configuration, however we are confident that testing the top two

Table 3: The overall ranking of heuristic selection (H.S.) and reward scheme (R.S.) pairs combined with top three sequence of low-level (meta)heuristics

| Examination | | | Post Enrolment | | | Curriculum-based | | |
|---|---|---|---|---|---|---|---|---|
| R.S.-H.S. | Order | Score | R.S.-H.S. | Order | Score | R.S.-H.S. | Order | Score |
| DE-TO | GD | 24.20 | DE-TO | GD-SA | 22.61 | NL-RW | GD | 38.66 |
| NL-TO | GD | 37.33 | DE-TO | GD | 37.48 | NL-TO | SA-GD | 39.56 |
| LI-RW | GD | 42.38 | DE-RW | GD | 55.17 | LI-RW | GD | 44.09 |
| NL-RW | GD | 42.77 | NL-TO | GD | 57.34 | LI-RW | SA-GD | 54.99 |
| LI-TO | GD | 52.99 | NL-TO | GD-SA | 58.36 | LI-TO | SA-GD | 55.51 |
| DE-RW | GD | 57.20 | LI-TO | GD-SA | 58.42 | DE-RW | SA–GD | 58.17 |
| LI-RW | HC-GD-SA | 59.70 | DE-RW | GD-SA | 63.59 | DE-RW | GD | 60.62 |
| DE-RW | HC-GD-SA | 79.09 | LI-TO | GD | 65.94 | NL-RW | SA-GD | 66.44 |
| NL-TO | HC-GD-SA | 79.75 | LI-RW | GD | 66.71 | NL-TO | GD | 66.79 |
| NL-RW | HC-GD-SA | 80.08 | NL-RW | GD | 74.34 | DE-TO | SA-GD | 68.47 |
| DE-TO | HC-GD-SA | 80.78 | NL-RW | GD-SA | 81.55 | DE-TO | GD | 74.08 |
| LI-TO | HC-GD-SA | 89.75 | LI-RW | GD-SA | 84.49 | LI-TO | GD | 98.61 |

ranked move acceptance orderings for each track is sufficient to produce the optimal configurations. Finally, we can see from the table that a non-linear reward mechanism with a Roulette Wheel selection and Great Deluge acceptance is the optimal configuration for the Curriculum-based track.

## 5 Experimental Results

Evaluation of the sequence-based hyper-heuristic approach was done using the optimised configuration obtained above. We ran each data instance supplied by the ITC for each track, that is: 8 instances for the Examination track, 16 instances for the Post Enrolment track and 21 instances for the Curriculum-based track. To accurately and fairly compare the proposed hyper-heuristic to the ITC competitors results all runs were performed under the same conditions. That is, each instance was evaluated by performing 10 complete runs, with a random seed, for the time allocated by the benchmarking program issued by the ITC. All runs were performed using the same windows machine as described in Section 4. The results described in Tables 4, 5 and 6 present our experimental results; the average and best for the Sequence-based Hyper-heuristic (SBHH). The tables also display the best scores of the competition winners alongside the best solution value produced by Thomàš Müller's code over 100 runs on each instance.

### 5.1 Examination Timetabling Results

Table 4 displays the results described above along with the best results of two bespoke approaches proposed after the competition finished. McCollum et al. [5] performed their approach for a total of 51 runs per instance. Saber et al. [28] performed their approach for a total of 21 runs per instance. The best scores for each track are displayed in bold. It is worth noting that the approaches in [1,27, 28] are all single stage approaches, generating a single solution without further

Table 4: Examination track experimental results, where entries in bold style highlight the best performing algorithms

| Instance | SBHH | | ITC-2007 Finalist | | | | | | Post ITC-2007 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Avg. | Best | Müller | [13] | [1] | [11] | [27] | [23] 100 runs | [5] 51 runs | [28] 20 runs |
| Exam 1 | 4074.8 | **4008** | 4370 | 5905 | 8006 | 6670 | 12035 | 4356 | 4633 | 6234 |
| Exam 2 | 391 | **385** | 400 | 1008 | 3470 | 623 | 3074 | 390 | 405 | 395 |
| Exam 3 | 10060.9 | 9347 | 10049 | 13862 | 18622 | - | 15917 | 9568 | **9064** | 13002 |
| Exam 4 | 19454.6 | 15870 | 18141 | 18674 | 22559 | - | 23582 | 16591 | **15663** | 17940 |
| Exam 5 | 2758 | **2617** | 2988 | 4139 | 4714 | 3847 | 6860 | 2941 | 3042 | 3900 |
| Exam 6 | 26867 | 26195 | 26950 | 27640 | 29155 | 27815 | 32250 | **25775** | 25880 | 27000 |
| Exam 7 | 3978.8 | **3824** | 4213 | 6683 | 10473 | 5420 | 17666 | 4088 | 4037 | 6214 |
| Exam 8 | 7228 | **7012** | 7861 | 10521 | 14317 | - | 16184 | 7565 | 7461 | 8552 |

improvement. Therefore, it is not completely fair to compare our results with these methods. However, our approach outperformed all other methods on 5 of the data instances. We also managed to provide better solutions than Müller's best over 100 runs on 7 of the data instances and outperformed Müller's ITC 2007 finalist solution values on all 8 instances. This displays the capability of the sequence-based hyper-heuristic for the Examination track. It would be interesting to perform further runs on each data instance to see if our method is capable of outperforming McCollum on instances 3 and 4, and Müller's best solution value achieved on instance 6 over 100 runs.

## 5.2 Post Enrolment-based Course Timetabling Results

Table 5 presents our experiments results for the Post Enrolment track of the competition. The results are displayed in a tuple, with the first number in each cell being the distance to feasibility (dtf) and the second being the overall solution value. We also compared our results to the best results obtained from state-of-the-art approaches developed after the competition [7,20,14,30,31]. The best scores for each track are displayed in bold.

The state-of-the-art approaches have made significant improvement within the problem of post enrolment with Ceschia et al. [7] displaying 12 of the best solution values making use of a single-step metaheuristic approach based on simulated annealing. Whilst our proposed technique only manages to obtain the joint best values for two of the instances, we managed to improve the solution value on 9 of Müller's competition runs. This indicates that our proposed method can produce improving solution compared to Müller's original solver, but it is not the best technique for solving Post Enrolment problems.

## 5.3 Curriculum-based Course Timetabling Results

Table 6 presents our experiments results for the final track, Curriculum-based timetabling. We compared our results to 4 approaches developed after the end of the competition: Tabu search and Iterative Local Search both developed

Table 5: Post Enrolment track experimental results, where entries in bold style highlight the best performing algorithms

| Ins. | SBHH | | ITC-2007 Finalist | | | | | | Post ITC-2007 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg. | Best | Müller | [6] | [1] | [8] | [25] | [23] 100 runs | [7] 30 runs | [20] | [14] | [30] | [31] |
| 1 | 153, 2259 | 0, 1810 | 0, 1861 | 0, 571 | 0, 61 | 0, 1482 | **0, 15** | 0, 1330 | 0, 59 | 0, 1166 | 0, 501 | 0, 650 | 0, 630 |
| 2 | 417, 2229 | 119, 2233 | 39, 2174 | 0, 993 | 0, 547 | 0, 1635 | **0, 0** | 0, 2154 | **0, 0** | 0, 1665 | 0, 342 | 0, 470 | 0, 450 |
| 3 | 0, 292 | 0, 234 | 0, 272 | 0, 164 | 0, 382 | 0, 288 | 0, 391 | 0, 205 | **0, 148** | 0, 251 | 0, 3770 | 0, 290 | 0, 300 |
| 4 | 0, 462 | 0, 386 | 0, 425 | 0, 310 | 0, 529 | 0, 385 | 0, 239 | 0, 394 | **0, 25** | 0, 424 | 0, 234 | 0, 600 | 0, 602 |
| 5 | 0, 42 | 0, 9 | 0, 8 | 0, 5 | 0, 5 | 0, 559 | 0, 34 | **0, 0** | **0, 0** | 0, 47 | **0, 0** | 0, 35 | 0, 6 |
| 6 | 0, 236 | 0, 76 | 0, 28 | **0, 0** | **0, 0** | 0, 851 | 0, 87 | 0, 13 | **0, 0** | 0, 412 | **0, 0** | 0, 20 | **0, 0** |
| 7 | 0, 20 | 0, 5 | 0, 13 | 0, 6 | **0, 0** | 0, 10 | **0, 0** | 0, 5 | **0, 0** | 0, 6 | **0, 0** | 0, 30 | **0, 0** |
| 8 | 0, 20 | **0, 0** | 0, 6 | **0, 0** | **0, 0** | **0, 0** | 0, 4 | **0, 0** | **0, 0** | 0, 65 | **0, 0** | **0, 0** | **0, 0** |
| 9 | 922, 2103 | 433, 2351 | 162, 2733 | 0, 1560 | **0, 0** | 0, 1947 | **0, 0** | 0, 1895 | **0, 0** | 0, 1819 | 0, 989 | 0, 630 | 0, 640 |
| 10 | 773, 2365 | 490, 2280 | 161, 2697 | 0, 2136 | **0, 0** | 0, 1741 | **0, 0** | 57, 2440 | 0, 3 | 0, 2091 | 0, 499 | 0, 2349 | 0, 663 |
| 11 | 0, 635 | 0, 437 | 0, 263 | 0, 178 | 0, 548 | 0, 240 | 0, 547 | 0, 347 | **0, 142** | 0, 288 | 0, 246 | 0, 350 | 0, 344 |
| 12 | 0, 825 | 0, 698 | 0, 804 | 0, 146 | 0, 869 | 0, 475 | **0, 32** | 0, 453 | 0, 267 | 0, 474 | 0, 172 | 0, 480 | 0, 198 |
| 13 | 0, 608 | 0, 302 | 0, 285 | **0, 0** | **0, 0** | 0, 675 | 0, 166 | 0, 74 | 0, 1 | 0, 298 | **0, 0** | 0, 46 | **0, 0** |
| 14 | 0, 545 | 0, 82 | 0, 110 | 0, 1 | **0, 0** | 0, 864 | **0, 0** | 0, 2 | **0, 0** | 0, 127 | **0, 0** | 0, 80 | 0, 35 |
| 15 | 0, 244 | **0, 0** | 0, 5 | **0, 0** | 0, 379 | **0, 0** | **0, 0** | **0, 0** | **0, 0** | 0, 108 | **0, 0** | **0, 0** | **0, 0** |
| 16 | 0, 167 | 0, 67 | 0, 132 | 0, 2 | 0, 191 | 0, 1 | 0, 41 | 0, 6 | **0, 0** | 0, 138 | **0, 0** | **0, 0** | 0, 140 |

Table 6: Curriculum-based track experimental results, where U indicates undefined and entries in bold style highlight the best performing algorithms

| Ins. | SBHH | | ITC-2007 Finalist | | | | | | | Post ITC-2007 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg. | Best | Müller | [22] | [1] | [16] | [9] | [23] 100 runs | [16] 30 runs | [22] TS | [22] ILS | [2] | [19] |
| 1 | 5 | **5** | **5** | **5** | **5** | **5** | 10 | **5** | **5** | **5** | **5** | **5** | 13 |
| 2 | 68 | 48 | 51 | 55 | 50 | 111 | 111 | **43** | 91 | 55 | 48 | 75 | **43** |
| 3 | 86 | 76 | 84 | **71** | 82 | 128 | 119 | 72 | 108 | 90 | 76 | 93 | 76 |
| 4 | 42 | **35** | 37 | 43 | **35** | 72 | 72 | **35** | 53 | 45 | 41 | 45 | 38 |
| 5 | 336 | 309 | 330 | 309 | 312 | 410 | 426 | 298 | 359 | 315 | **303** | 326 | 314 |
| 6 | 60 | **35** | 48 | 53 | 69 | 102 | 130 | 41 | 79 | 58 | 54 | 62 | 41 |
| 7 | 26 | **10** | 20 | 28 | 42 | 57 | 110 | 14 | 36 | 33 | 25 | 38 | 19 |
| 8 | 43 | **39** | 41 | 49 | 40 | 77 | 83 | **39** | 63 | 49 | 47 | 50 | 43 |
| 9 | 108 | **101** | 109 | 105 | 110 | 150 | 139 | 103 | 128 | 109 | 106 | 119 | 102 |
| 10 | 25 | 11 | 16 | 21 | 27 | 71 | 85 | **9** | 49 | 23 | 23 | 27 | 14 |
| 11 | 0 | **0** | **0** | **0** | **0** | **0** | 3 | **0** | **0** | **0** | **0** | **0** | **0** |
| 12 | 353 | 333 | 333 | 343 | 351 | 442 | 408 | 331 | 389 | 330 | **324** | 358 | 405 |
| 13 | 78 | **59** | 66 | 73 | 68 | 622 | 113 | 66 | 91 | 71 | 68 | 77 | 68 |
| 14 | 60 | 55 | 59 | 57 | 59 | 90 | 84 | **53** | 81 | 55 | **53** | 59 | 54 |
| 15 | 85 | 75 | 84 | **71** | 82 | 128 | 119 | U | U | U | U | 87 | U |
| 16 | 45 | 38 | **34** | 39 | 40 | 81 | 84 | U | U | U | U | 47 | U |
| 17 | 88 | **75** | 83 | 91 | 102 | 124 | 152 | U | U | U | U | 86 | U |
| 18 | 84 | 75 | 83 | 69 | **68** | 116 | 110 | U | U | U | U | 71 | U |
| 19 | 71 | 64 | **62** | 65 | 75 | 107 | 111 | U | U | U | U | 74 | U |
| 20 | 53 | 34 | **27** | 47 | 61 | 88 | 144 | U | U | U | U | 54 | U |
| 21 | 116 | **100** | 103 | 106 | 123 | 174 | 169 | U | U | U | U | 117 | U |

in [22, 2, 19]. We were also able to compare our solutions to the best solution values obtained by Geiger [16] over 30 runs for each instance, with each run allowing 100,000,000 evaluations. Solution values are not available for all instances for each approach. The solution values we were not able to obtain are represented with a 'U' for unknown and all the best values for a given instance are displayed in bold. Our approached achieved 10 of the best solution values across all 21 instances. We managed to make improvements on Müller's competition final solutions across 15 of the instances. We also made improvements on 4 instances compared to Müller's 100 runs.

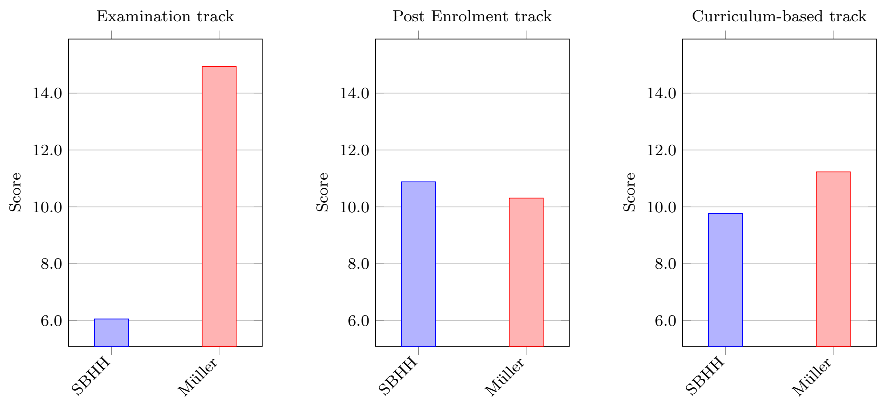

Fig. 3: Overall performance comparison of Müller's approach and SBHH based on average ranking scores for the Examination, Post Enrolment and Curriculum-based tracks of ITC 2007

### 5.4 Performance Comparison to Müller's Approach

In addition to comparing the best solution values produced by the proposed and various other approaches, we have calculated the average ranking scores for each track based on Müller's and our solutions obtained for ITC 2007. We calculated the ranking scores in the same manner as the ITC calculated the winner, and similarly we ranked the different heuristics in the configuration results section. Figure 3 displays the average ranking scores for our and Müller's approaches for each timetabling track of ITC 2007. The figure shows that our approach outperformed Müller's on the Examination track and Curriculum-based track by a significant margin producing average ranking scores of 6.06 and 9.77 versus 14.94 and 11.23, respectively. This figure also displays that Müller's original solver performs marginally better than with the sequence-based hyper-heuristic implemented with an average ranking score of 10.31 against 10.88. However, we noticed that the proposed implementation had difficulties with data instances that often resulted with a dtf (distance to feasibility) score. If we exclude such Post Enrolment track instances, including 2, 9, 10, the overall average ranking of our approach becomes 10.16 against Müller's approach 11.07, hence we can observe that the sequence-based hyper-heuristic performed better on the remaining 13 instances.

## 6 Conclusions

There has been a growing body of work on multi-stage selections hyper-heuristics for both single [18] and multi-objective [21] optimisation. This paper presents a tuned multi-stage sequence-based hyper-heuristic approach embedded into the CPSolver framework in an attempt to improve on the solutions produced by Müller in the ITC 2007 competition. The proposed method is capable of improving an initially created solution. It does this in a more sophisticated manner than the original framework by attempting to adaptively identify and isolate sequences of low-level heuristics that perform well on a given instance. We carried out extensive testing to accurately determine the best configuration setup for our approach to decide the best stage option (acceptance ordering), heuristic selection method and reward scheme. The following stage options are considered: HC-GD-SA, HC-SA-GD, GD-SA, SA-GD, GD and SA. We observed that Great Deluge operating by itself ranked best for all three tracks. However, when testing the heuristic selection method and reward mechanism we found that the performance of the acceptance strategy is not independent of other factors. After configuration tests, we concluded that the best configurations for the Examination, post enrolment, and Curriculum-based tracks were Great Deluge with tournament selection and delta learning, Great Deluge and Simulated Annealing with tournament selection and delta learning, and Great Deluge with Roulette Wheel selection and non-linear learning, respectively.

As illustrated, the proposed approach was effective at improving on Müller's ITC 2007 solution objective values. All the Examination instances were improved upon, with instance 4 improving as much as 12.5%. We improved on the solution values for 9 data instances in the Post Enrolment track. Finally, for the curriculum-based track we made improvements on 15 of the data instances, as much as 50% (instance 7), and found the same solution value on 3 other data instances. We also compared the proposed technique against state-of-the-art approaches for all three tracks. We found improving solution values on 5 Examination data instances, joint lowest solution values on 2 post enrolment data instances and finally, 6 improving and 4 joint lowest solution values on the curriculum-based track. It is stressed that the results obtained by methods after the end of the competition are bespoke solvers designed to only solve problem instances for the given track. Some of the solvers were also allowed to run for more than 10 runs, the allowed number within the competition, leading to an unfair comparison. The educational timetabling is still of interest to many academics as well as practitioners. A trivial future work would be applying the proposed approach to unseen instances, perhaps to those instances provided during the new competition on university course timetabling, ITC 2019 [24].

## References

1. Atsuta, M., Nonobe, K., Ibaraki, T.: Itc-2007 track2: An approach using general csp solver (2007)
2. Bonutti, A., De Cesco, F., Di Gaspero, L., Schaerf, A.: Benchmarking curriculum-based course timetabling: Formulations, data formats, instances, validation, visualization, and results. Annals OR **194**, 59–70 (2012)
3. Burke, E., Newall, J.: Solving examination timetabling problems through adaption of heuristic orderings: Models and algorithms for planning and scheduling problems (edi-

tors: Philippe baptiste, jacques carlier, alix munier, andreas s. schulz). Annals of Operations Research **129** (2004)
4. Burke, E.K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Qu, R.: Hyper-heuristics: a survey of the state of the art. Journal of the Operational Research Society **64**(12), 1695–1724 (2013)
5. C Mccollum, G., J Mcmullan, P., Parkes, A., K Burke, E., Abdullah, S., Mccollum, B.: An extended great deluge approach to the examination timetabling problem (2019)
6. Cambazard, H., Hebrard, E., O'Sullivan, B., Papadopoulos, A.: Local search and constraint programming for the post enrolment-based course timetabling problem. Annals of Operations Research **194**, 111–135 (2012)
7. Ceschia, S., Di Gaspero, L., Schaerf, A.: Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. Computers & Operations Research **39** (2011)
8. Chiarandini, M., Fawcett, C., Hoos, H.H.: A modular multiphase heuristic solver for post enrolment course timetabling (2008)
9. Clark, M., Henz, M., Love, B.: Quikfix—a repair-based timetable solver. In: Proceedings of the Seventh International Conference on the Practice and Theory of Automated Timetabling
10. Cowling, P., Kendall, G., Soubeiga, E.: A hyperheuristic approach to scheduling a sales summit. In: International Conference on the Practice and Theory of Automated Timetabling, pp. 176–190. Springer (2000)
11. De Smet, G.: Itc2007 - examination track. p. 22 (2008)
12. Drake, J.H., Kheiri, A., Özcan, E., Burke, E.K.: Recent advances in selection hyper-heuristics. European Journal of Operational Research (2019). DOI https://doi.org/10.1016/j.ejor.2019.07.073
13. Gogos, C., Alefragis, P., Housos, E.: A multi-staged algorithmic process for the solution of the examination timetabling problem, proceedings of the international conference on practice and theory of automated timetabling (patat 2008). p. 22 (2008)
14. Jat, S.N., Yang, S.: A hybrid genetic algorithm and tabu search approach for post enrolment course timetabling. J. Scheduling **14**, 617–637 (2011)
15. Jebari, K.: Selection methods for genetic algorithms. International Journal of Emerging Sciences **3**, 333–344 (2013)
16. Josef Geiger, M.: Applying the threshold accepting metaheuristic to curriculum based course timetabling. Annals of Operations Research **194**, 189–202 (2008)
17. Kheiri, A., Keedwell, E.: A hidden markov model approach to the problem of heuristic selection in hyper-heuristics with a case study in high school timetabling problems. Evol. Comput. **25**(3), 473–501 (2017)
18. Kheiri, A., Özcan, E.: An iterated multi-stage selection hyper-heuristic. European Journal of Operational Research **250**(1), 77 – 90 (2016)
19. Lach, G., Lübbecke, M.: Curriculum based course timetabling: New solutions to udine benchmark instances. Annals of Operations Research **194**, 255–272 (2012)
20. Lewis, R.: A time-dependent metaheuristic algorithm for post enrolment-based course timetabling (2012)
21. Li, W., Özcan, E., John, R.: A learning automata-based multiobjective hyper-heuristic. IEEE Transactions on Evolutionary Computation **23**(1), 59–73 (2019). DOI 10.1109/TEVC.2017.2785346
22. Lü, Z., Hao, J.K.: Solving the course timetabling problem with a hybrid heuristic algorithm. pp. 262–273 (2008)
23. Müller, T.: Itc2007 solver description: A hybrid approach. Annals of Operations Research **172**, 429–446 (2008). DOI 10.1007/s10479-009-0644-y
24. Müller T, R.H., Müllerova, Z.: University course timetabling and international timetabling competition 2019, proceedings of the 12th international conference on the practice and theory of automated timetabling (patat 2018). p. 27 (2018)
25. Nothegger, C., Mayer, A., M. Chwatal, A., Raidl, G.: Solving the post enrolment course timetabling problem by ant colony optimization. Annals OR **194**, 325–339 (2012)
26. Özcan, E., Bilgin, B., Korkmaz, E.: A comprehensive analysis of hyper-heuristics. Intelligent Data Analysis **12**, 3–23 (2008). DOI 10.3233/IDA-2008-12102

27. Pillay, N.: A developmental approach to the examination timetabling problem, proceedings of the international conference on practice and theory of automated timetabling (patat 2008). pp. 19–22 (2008)
28. Sabar, N., Ayob, M., Qu, R., Kendall, G.: A graph coloring constructive hyper-heuristic for examination timetabling problems. Applied Intelligence - APIN **37**, 1–11 (2011). DOI 10.1007/s10489-011-0309-9
29. Sörensen, K., Glover, F.W.: Metaheuristics, pp. 960–970. Springer US, Boston, MA (2013)
30. Soria-Alcaraz, J., Ochoa, G., Swan, J., Carpio, M., Soberanes, H., K. Burke, E.: Effective learning hyper-heuristics for the course timetabling problem. European Journal of Operational Research **238**, 77–86 (2014)
31. Soria-Alcaraz, J., Özcan, E., Swan, J., Kendall, G., Carpio, M.: Iterated local search using an add and delete hyper-heuristic for university course timetabling. Applied Soft Computing **40** (2015)