Open Access **Baghdad Science Journal** **P-ISSN: 2078-8665**
2021, 18(4) Supplement: 1465-1475 **E-ISSN: 2411-7986**

DOI: https://dx.doi.org/10.21123/bsj.2021.18.4(Suppl.).1465

Effective Solution of University Course Timetabling using Particle Swarm Optimizer based Hyper Heuristic approach

Zahid Iqbal [1,2 *] ***Rafia Ilyas*** [2] ***Huah Yong Chan*** [1] ***Naveed Ahmed*** [2]

[1]School of Computer Sciences, Universiti Sains Malaysia, 11800, Pulau Pinang, Malaysia.
[2]Department of Computer Science, University of Gujrat, Gujrat, Pakistan.
[*]Corresponding author: zahidiqbal.mb@gmail.com
E-mails: rafy.choudary@gmail.com , hychan@usm.my , naveed49478@gmail.com

Received 18/10/2021, Accepted 14/11/2021, Published 20/12/2021

Abstract:

The university course timetable problem (UCTP) is typically a combinatorial optimization problem. Manually achieving a useful timetable requires many days of effort, and the results are still unsatisfactory. unsatisfactory. Various states of art methods (heuristic, meta-heuristic) are used to satisfactorily solve UCTP. However, these approaches typically represent the instance-specific solutions. The hyper-heuristic framework adequately addresses this complex problem. This research proposed Particle Swarm Optimizer-based Hyper Heuristic (HH PSO) to solve UCTP efficiently. PSO is used as a higher-level method that selects low-level heuristics (LLH) sequence which further generates an optimal solution. The proposed approach generates solutions into two phases (initial and improvement). A new LLH named "least possible rooms left" has been developed and proposed to schedule events. Both datasets of international timetabling competition (ITC) i.e., ITC 2002 and ITC 2007 are used to evaluate the proposed method. Experimental results indicate that the proposed low-level heuristic helps to schedule events at the initial stage. When compared with other LLH's, the proposed LLH schedule more events for 14 and 15 data instances out of 24 and 20 data instances of ITC 2002 and ITC 2007, respectively. The experimental study shows that HH PSO gets a lower soft constraint violation rate on seven and six data instances of ITC 2007 and ITC 2002, respectively. This research has concluded the proposed LLH can get a feasible solution if prioritized.

Keywords: Auto Timetable, Hyper Heuristic, Particle Swarm Optimizer.

Introduction:

In this progressive era, timetabling remains a critical problem that is experienced in many forms. The versatile nature of the timetabling problem makes it among challenging issues for artificial intelligence and operational research. Among all timetabling issues, educational timetabling is one of the most complicated tasks and is widely discussed by researchers. Generally, there are three types of educational timetabling (university timetable, exam timetable, and school timetable). University course timetabling problem (UCTP) is a regularly experienced issue.

Therefore, UCTP has been debated internationally for the last few decades, and its significance has increased. This is because the manual solution to the timetabling problem is very time-consuming as all requirements need to be taken care of. But, still, the results are not satisfactory [1]. UCTP has two categories of constraints: hard constraints and soft constraints [2]. A solution timetable is feasible only if no hard constraint is violated and schedule all events. Because of the complex nature of the timetabling problem, usually, it might be impossible to approach a solution with no soft constraint violation [3].

It has been explained that all timetable problems are NP-Complete [4]. Even though several heuristic and metaheuristic methods exist but most of these are problem-specific. Moreover, heuristic methods are expensive to develop and maintain. It is needed to work on more generalized methods that can be used for more than one problem instance or even for more than one problem domain.

Hyper heuristics have great potential to produce more general solutions. However, there has been relatively little research on hyper-heuristic for

UCTP. Research indicates that Particle Swarm Optimizer (PSO) is capable of handling combinatorial optimization problems and constrained optimization problems [2,5].

PSO has widely been used in the literature[6–9]. This research proposed a PSO-based selection hyper-heuristic for solving UCTP where PSO will work as a higher-level methodology that will use low-level heuristics to obtain the optimal solution. Moreover, the role of low-level heuristics is noted. Furthermore, a new low-level heuristic is proposed that helps to achieve a feasible solution. The proposed methodology is tested on both problem formulation given in the first and second international timetabling competitions (ITC) 2002 and 2007 and results are compared with the previous state of art methods.

Problem Representation:

UCTP can be formulated as Constraint Satisfaction Problem. There are multiple ways to represent UCTP. It is almost beyond the possibility to compose a single formulation that fits in for all universities. As each university has various rules, resources, and costs. This research employed a formulation for UCTP named "ITC 2007" that was proposed in International Timetabling Competition (ITC) [10].

The core objective is to insert all events or classes in the solution timetable (i.e., assign room and timeslots to each event from a limited collection of rooms and timeslots with appropriate resources in the rooms), such that every event must obey five hard constraints. There are five hard constraints used in ITC 2007. H1. Conflict (not more than one class scheduled at a time for one student.), H2. Compatibility (the room that is used for scheduling an event must have enough space and resources.), H3. Occupancy (in one timeslot, only one class can be scheduled in one room.), H4. Availability (an event can only be scheduled in a timeslot from the collection of possible timeslots given for that event.), H5. Precedence (events may or may not have precedence on each other if precedence is given it must be followed.). The resultant timetable can be acceptable if it is valid or feasible. A timetable can only be valid when no hard constraint is violated, but some events remain unplaced in the solution. A timetable is feasible if all events are scheduled in solution with no violation of hard constraints. Also, three soft constraints used in ITC 2007 are S1. Late Events (avoid scheduling events in the last timeslot), S2. Consecutive Events (no consecutive events (three or more) to attend by any student in a day), S3. Isolated Events (each student should have more than one event to attend in a day).

Distance to Feasibility measure is used to evaluate the solution. If a solution is valid, it is computed by counting the number of students who must have to attend "unplaced events". For example, two events left unplaced in a solution. And students attending these events are 15 and 8. Then Distance to Feasibility will be (15 + 8) = 23. When a solution is feasible then, the Distance to Feasibility will be zero. This will happen only when there is no hard constraint violation (HCV) in the solution. After calculating Distance to Feasibility, soft constraint violations (SCV) are calculated in the objective function that counts students (that attend only one event in a day, have three or more consecutive classes, or have class in the last timeslot). Classes in two consecutive days (i.e., at the end of one day and the start of the second day) will not be counted as consecutive. Table I shows different formulations and constraints used in those formulations along with data instances.

Table 1. Constraints and Data Instances

Problem Set	H1.	H2.	H3.	H4.	H5.	S1.	S2.	S3.	Number of Data Instances	Years
ITC 2007	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	24	2007
ITC 2002	Yes	Yes	Yes	No	No	Yes	Yes	Yes	20	2002

Literature Review:

In the past few years, several heuristics, metaheuristic, and hyper-heuristic methods have been used for UCTP. The heuristic is applied to find out near-optimal solutions. It is based on a random search mechanism. Although heuristic does not promise the globally best solution, still it is beneficial when traditional methods fail.

Metaheuristics study the high-level methods designed from heuristics. It is an iterative generation process that guides a subordinate heuristic by combining intelligently various concepts for exploring, exploiting the search space, and learning strategies are used to structure information for finding a near-optimal solution [11]. There are two groups of metaheuristics methods [12]: methods based on *Local search* and methods based on population. Local Search-based methods deal with a single object in one iteration. Local search-based techniques are also called single solution-

based techniques. In these approaches, a single solution is created, then modified using local search. Multi-Neighborhood [13], Tabu Search (TS) [14,15], Simulated Annealing (SA) [16], Local Search (LS) [17] are some local search-based methods. Population search deals with the population of objects on each iteration. Multiple Solutions are built to explore search space and move toward an optimal solution. Ant Colony algorithm [18], Particle Swarm Optimization (PSO) [9], and Genetic algorithm (GA) [19] are some population-based approaches. Several metaheuristic methods have been used for UCTP [13–16,18].

A blend of metaheuristic methods [13] was proposed for UCTP to solve the problem set of ITC 2007. This approach used neighborhood search to find a feasible solution at first. After that simulated annealing was used to minimize the conflicts. Exchange of timeslots and rooms of scheduled events was performed to decrease the conflicts. Finally, the event was scheduled to the timeslot that caused comparatively fewer conflicts. In the second step, tabu search was used to add randomness in that approach. A general-purpose constraint satisfaction solver using tabu search [14] was tested on ITC 2007 of UCTP. This approach used weighted constraints to track conflicts in the solution timetable. It dynamically controlled the tabu tenure and adjusted the weight of the constraint to obtain a feasible solution. A multiphase heuristic solver was proposed by [15]. In the first phase, hard constraints were considered only. At the end of the first phase, there might be some events that remain unscheduled. In the second phase, tabu search was used to schedule unplaced events. The best partial solution was used side by side for minimizing soft constraint violation after a specified number of non-improving moves. Iterated local search was used several times with the blend of other techniques i-e. simulated annealing [17], great deluge [16], and hill-climbing to produce feasible solutions for UCTP. An ant colony algorithm was proposed by [18] for UCTP. Ants were used as events that were further scheduled in timeslots and rooms. Events were scheduled randomly using the greedy approach. This method was also examined on ITC 2007.

Hyper Heuristic is a generalized search technique that uses “heuristic for selecting heuristics” or “heuristics to generate heuristics”. It does not search solution space directly but searches for a method or set of heuristics that can solve problem instances [20]. By definition, it is clear HH can be classified into two types: generation hyper-heuristics and selection hyper-heuristics. Hyper-heuristic is just two decades older. It was presented there as a higher-level method, which could choose or generate low-level heuristics at each decision point. Hyper-heuristic was first proposed for UCTP in 2003. An evolutionary algorithm (EA) is used as a high-level methodology. One-point cross-over, binary tournament selection, and mutation are used by this algorithm. Performance is evaluated on ITC 2002 dataset [21]. In 2004-05, messy genetic algorithms were proposed for UCTP [19,22]. These algorithms are based on selection constructive hyper-heuristic, where messy Genetic Algorithm (GA) is used as a high-level methodology to explore heuristic space. In 2007, Simulated Annealing (SA) was used for UCTP that uses random selection of heuristic in start until a history of heuristics performance is gathered. This history is further used in solution generation [23]. Using multiple high-level heuristics improved the quality of the solution. In 2010, a random greedy approach based on generation perturbative hyper-heuristic was proposed [24]. This approach initially works on random search and then improves the solution by using perturbative heuristics. In 2014, iterated local search-based hyper-heuristic was used for UCTP, which reported the best results on ITC 2007 data set [25]. In 2016, add delete hyper-heuristic was proposed, it added or deleted events from solution using iterated local search to improve the solution [26]. It was assessed on ITC 2002 dataset and produced comparable results. Research shows that Particle Swarm Optimizer (PSO) [27] works better than the genetic algorithm for UCTP. PSO is a member of the broader swarm intelligence field for solving global optimization problems. A hybrid PSO combined with local search is evaluated for university course timetabling problems [28]. The result shows that hybrid PSO performs better than PSO alone and genetic algorithms. PSO was used for UCTP with a flexible problem set. The proposed method was tested on the dataset of the University of Taiwan [7]. In 2013, another hybrid approach for PSO was proposed by [29]. Transition probability was applied at the place of velocity. Results showed that this hybrid approach produced a lesser number of conflicts with small data set but performance level falls on larger data sets. However, it performed better than other evolutionary approaches. This method worked effectively for the university of Tsukuba dataset. In 2014**,** a comparative study was conducted, which proved that PSO can solve university lecture timetabling problems better than genetic algorithms [30]. PSO gets one near the optima faster than GA, but GA eventually gets one closer. Many researchers believe that getting a near-optimal solution faster is better than achieving an optimal solution in timetabling problems. Recently, PSO has been used for school timetabling problems

[9] and produced reasonable results. Some remarkable works on UCTP have been done by [13–16,18] using different heuristic, metaheuristic [31], and hyper-heuristic methods. Authors [32] have given a comprehensive review of hyper-heuristic methods, nature of heuristic space with data sets used in hyper-heuristic models. Although research on meta-heuristic [33–35] and hyper-heuristic methods are going side by side however when it comes to flexible methods, that can work for different problem formulations, the hyper-heuristic framework has great potential to produce general solution. In past, few years, PSO based hyper-heuristics is successfully used to solve problems in different domains. But only a few hyper-heuristic methods are proposed for UCTP. Beneficial work can be done for UCTP by employing the power of hyper-heuristic.

Proposed Methodology:

This research proposed a PSO-based selection hyper-heuristic (HH-PSO) for the university course timetabling issue. To the best of our knowledge and survey, the PSO-based selection hyper-heuristic is yet unused for university course timetabling issues. PSO has a collection of individuals named particles. It updates each particle's movement, in its population, known as swarm, instead of generating a new population. The position of every particle is updated based on its own previous best position and the global best position of its neighbor particles. PSO works on the food searching strategy used by birds. An individual bird in a crowd act as a particle and a swarm is a group of particles. Particles move into search space in different dimensions. Every particle saves two basic sets of information (global best, personal best). Global and personal best is retrieved based on fitness, calculated by objective function [27]. In PSO, each particle adjusts its best position in the guidance of its previous best position. And the global best is achieved by evaluating the best position of every particle on each iteration. Generally, PSO can be formulated as follows: The D dimensional search space, $X_i = X_{i1} + X_{i2} + X_{i3} + \cdots + X_{iD}$ is position of i^{th} particle $P_i = (P_{i1}, P_{i2}, P_{i3}, \dots, P_{id})$ is position of particle i in search space, $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$ is velocity of each particle and g is global best. Velocity and position of particles are updated by using following equations:

$$V_{id} = x * (w * V_{id} + c_1 * r_1 * (P_{id} - X_{id}) + c_2 * r_2 * (P_{gd} - X_{id}) \quad (1)$$

$$X_{id} = X_{id} + V_{id} \quad (2)$$

There are two random numbers used in the above equations presented as r1 and r2. The value of r_1 and r_2 is between the range (0,1). The two positive constants are named c1 and c2. To manage velocity magnitude x is used as a constriction factor. Proper mapping of the problem is very important to get useful results. In this research, PSO is used as a high-level methodology that will perform direct encoding of problem representation. More specifically, a complete solution is provided by each particle. A list of events is used that number corresponds with the timeslot and room for that room. Each particle represents a list of events that length is equivalent to the number of events in a dataset.

Then PSO selects low-level heuristics that will solve the timetabling problem. Both constructive and perturbative low-level heuristics will be used. Construction heuristics are used for estimating the difficulty level of an event in scheduling. Graph-based heuristics are construction heuristics. Different construction heuristics are [36] largest degree, largest enrolment, largest weighted. [3] perturbative LLH improves firstly created solution i.e., swapping two events, reallocating an event, and rescheduling an event, etc. Each particle in the population is completed timetable. The movement of particles depends on the objective function value, personal best (pbest), and global best (gbest) of the particle. Fig. 1 shows the basic algorithm of HH-PSO. It has two main processes: initialization and modification. To generate the initial solution the process of initialization works. The detailed algorithm of initialization is given in Fig. 2. After initial solution generation, the pbest and gbest of every particle are calculated. All the hard constraints are satisfied in the initial solution. There can be following four possible scenarios to handle constraints in solution generation: Infeasible/ Infeasible (initial/ modified), Infeasible/ Feasible (initial/ modified), Feasible/ Infeasible (initial/modified), Feasible/Feasible (initial/modified) As mentioned above that this research has set the focus on the feasible initial solution. Hence, only the 4th scenario is considered in this work. Particles with low pbest move towards gbest particle in the modification section. The modification process given in Fig. 3 improves the solution by reducing soft constraint violation as much as possible by randomly selecting events and swapping to other timeslots.

Input: parameter values
Output: feasible timetable
Algorithm: Hyper-Heuristic PSO

1. Generate initial solution
2. Set pbest and calculate gbest
3. Improve initial solution
4. Update gbest and pbest.
5. Check termination criteria if condition not met goto 3.

Figure 1. Basic Proposed Methodology HH-PSO

Input: data set, initial heuristic list {1, 2, 3, 4, 5,6, 7}
Output: feasible timetable
Algorithm: Initialization

1. Schedule events that have minimum possible rooms left in partial solution.
2. Schedule events that have least possible timeslots left in partial solution
3. Select a Random Sequence of Low-Level Heuristic (LLH)
4. If (sequence does not match the rejected sequences)
   For e =0 to e = number of events in dataset
   Schedule the events with respect to the sequence of LLH
   If a LLH schedule the event repeat it once again immediately
   Else follow the sequence.
5. Compute cost of solution: hard constraint violation (HCV)
   If the sequence fails to produce solution add it to rejected sequences list
   regenerate new sequence for solution generation
   Else
   accept the initial solution.

Figure 2. Initialization Process of HH-PSO

Input: initial solution
Output: feasible timetable
Algorithm: Improvement

1. Select a random event and updates its timeslot and room.
2. If hard constraint violation remains 0 and soft constraint violation improves or remain consistent
   Update the timeslot and room.
3. Else Don't change the solution.
4. If particle(i) does not improve in an iteration.

Replace particle(i) with gbest at that iteration.

Figure 3. Improvement process of proposed method (HH-PSO)

Initialization

In this research, intelligent initialization has been applied which generates a feasible initial solution. Seven low-level heuristics (LLH) are used for this purpose. Among these seven LLH, the first six are widely used in literature for UCTP [3]. The structure of these LLH helps to consider conflicts (hard constraint violation) during the assignment of timeslot and room to the events. While seventh is proposed in this research, based on an experiment calculated during result generation. We named it the *least possible rooms left*. During experiments, it was observed that events that have minimum possible timeslots or room left in partial solution might be left unplaced at the end. To adjust them, we used these two heuristics at the start. It helped us to first schedule the events that are hard to schedule. This approach helps to reduce the number of unplaced events in the solution without violating any hard constraint. We have used the following LLH's; Least Saturated Degree First (SD), Largest Enrolment (LE), Largest Color Degree First (LCD), Largest Weighted Degree First (LWD), Random Ordering (RO), Largest Degree First (LD) [3] and Least Possible Rooms Left (MRL). Fig. 2 shows the initial construction level algorithm. After scheduling the hard events, PSO selects a sequence of LLH from a predefined set of sequences and uses these LLH to schedule the remaining events. If any sequence fails to schedule events, that sequence is added to the rejected list. And next time new sequence is compared with rejected list sequences. The sequences in the rejected list are not used further for solution generation. This method saves time. On the other hand, if a heuristic successfully schedules an event it is repeated to schedule the next event too. It is expected that a set of the same heuristics helps schedule events. Random ordering is used to add randomness in solution construction. This initial method will generate and evaluate a feasible population of N size and it will update the pbest and gbest accordingly. By initiating every particle with a feasible solution, we hope that it will approach near to optimal solution and may bypass many local optimums. This is how premature convergence is avoided. After generating the initial solution, the performance of solution is evaluated.

The research used an objective function/ penalty function that is proposed in ITC 2007 competition with modification. In ITC 2007, two types of solutions were considered: valid solution and feasible solution [10]. This research focused on initial feasible solution generation, so we eliminate the valid solution section. And considered hard constraints violation only. This allows us to generate a solution that schedules all events while satisfying all hard constraints in lesser time. After completing the generation of the initial solution, control moves to the main algorithm given in Fig. 1.

It will further set the pbest and gbest based on the objective function.

Optimization:

The second step of the proposed method is to improve the quality of the initial solution. The mutation operator prevents fast convergence of the algorithm and presumably trapping in local minimums. Fig. 3 shows the optimization process in algorithmic form. For now, only one structure is used that "Randomly select an event and replace it on any new timeslot and room that must not cause any hard constraint violation and improves the soft constraint violation". Another action that is considered to control slow convergence is to replace the particle that may not improve after several iterations with the gbest at that iteration.

Results:

The results are divided into three groups i.e., the effectiveness of low-level heuristics, factors that affect the output, testing the methodology on standard datasets (ITC 2007, ITC 2002). The results are further compared with previously proposed techniques using the same data sets.

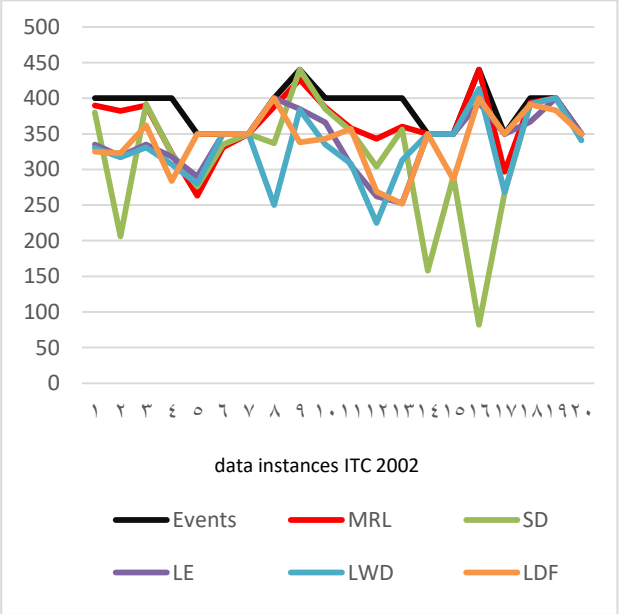

Figure 4. Number of events scheduled by each low-level heuristic on ITC 2002 data instances

This research used seven low-level heuristics. We have used a single heuristic on all dataset instances of ITC 2002 and ITC 2007 and calculated the number of events scheduled by a single heuristic without violating any hard constraint. It is observed that our proposed heuristic that is "min room" has scheduled more events with no hard constraint violation than other heuristics. It was previously proved in the literature that Least Saturated Degree solves more events. Our proposed heuristic named minimum possible room left (MRL) results better than Least Saturated Degree on ITC 2007 and ITC 2002 datasets. For ITC 2002, it scheduled more events of 14 datasets. For ITC 2007, it scheduled more events of 15 datasets instance. The result is shown in Fig. 4 and Fig. 5. Fig. 4 shows total events in ITC 2002 dataset instances and the number of events scheduled by each heuristic using datasets. Fig. 5 shows the same for ITC 2007 data set. It can be observed that overall, our proposed heuristic schedules more events. The least saturated degree solves 2nd most events for all instances. It is observed that Min room scheduled more events than other LLH in most cases with a lesser soft constraint violation.

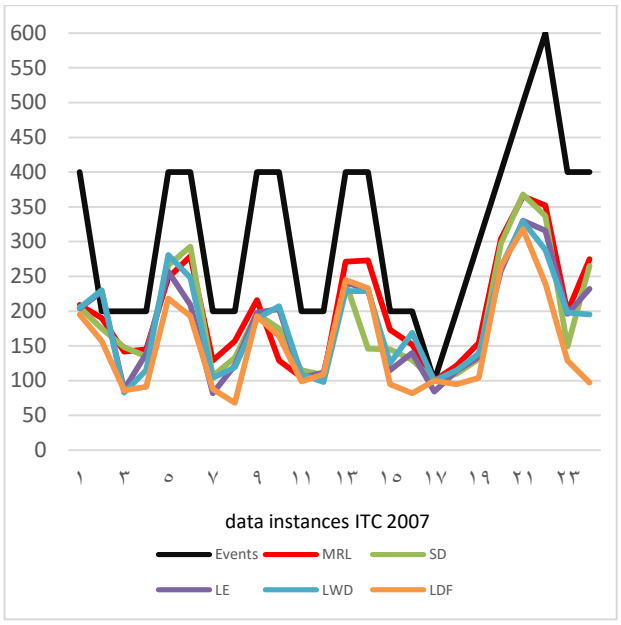

Figure 5. Number of events scheduled by each low-level heuristic on ITC 2007 data instances

Moreover, the effect of different parameters and their impact on output is analyzed. These factors are divided into two parts: dataset factors and algorithmic factors. Different dataset factors can increase/decrease the complexity of dataset instances. The values of these factors may affect the quality of the timetable. Table II holds the values of data sets i.e., number of students, number of events, number of rooms, and number of features required by events and may be held by rooms. It is observed from the experiments that events that require more room features may cause difficulty in improvement. As there may have less set of possible rooms for each event available in the partial timetable. Dataset instances that have more students and fewer events may have a greater number of conflicts involved. It may cause a problem in the improvement phase due to more conflict density.

Baghdad Science Journal
P-ISSN: 2078-8665
E-ISSN: 2411-7986

Table 2. Comparison of data instances of ITC 2002 and ITC 2007

Data	Number of events		Number of students		Number of rooms		Number of features	
data set	ITC 2002	ITC 2007	ITC 2002	ITC 2007	ITC 2002	ITC 2007	ITC 2002	ITC 2007
1	400	400	200	500	10	10	10	10
2	400	400	200	500	10	10	10	10
3	400	200	200	1000	10	20	10	10
4	400	200	300	1000	10	20	5	10
5	350	400	300	300	10	20	10	20
6	350	400	300	300	10	20	5	20
7	350	200	350	500	10	20	5	20
8	400	200	250	500	10	20	5	20
9	440	400	220	500	11	10	6	20
10	400	400	200	500	10	10	5	20
11	400	200	220	1000	10	10	6	10
12	400	200	200	1000	10	10	5	10
13	400	400	250	300	10	20	6	10
14	350	400	350	300	10	20	5	10
15	350	200	300	500	10	10	10	20
16	440	200	220	500	11	10	6	20
17	350	100	300	500	10	10	10	10
18	400	200	200	500	10	10	10	10
19	400	300	300	1000	10	10	5	10
20	350	400	300	1000	10	10	5	10
21	N/A	500	N/A	300	N/A	10	N/A	10
22	N/A	600	N/A	500	N/A	20	N/A	20
23	N/A	400	N/A	1000	N/A	20	N/A	30
24	N/A	400	N/A	1000	N/A	20	N/A	30

Algorithmic Factors:

In PSO, there are certain parameters. Their values have a greater effect on algorithm performance. Table 3 holds the values of parameters used in this research. Size of Swarm: The population size of the particle has a greater effect on the output of PSO. From experiments and previous studies, we have executed our algorithm on 5, 10, and 20 particles. It is observed that HH PSO produced better results on 20 particles.

Table 3. PSO Parameter Values used in Experiments

Attribute	Values
Initialization method	Supervised with random
Swarm Size	20
Max iteration	50, 100
Cognitive and social ratio	2:2
Inertia weight	0.75, 0.6
Data sets	ITC 2002 and ITC 2007

Number of Iterations

Exploration and exploitation of PSO can be affected by the number of iterations. We have used 50 and 100 iterations. Even on 50 iterations improvement in solution is achieved in this research.

Social and cognitive factors

Typical values of social and cognitive factors are used in this research.

Initialization

The iinitialization method plays a great role in the effective generation of solutions. This research uses a blend of supervised and random methodology that produces feasible solutions even on the initial level.

Comparison:

HH PSO results given in table 4 and table 5 are compared with some well-known previously proposed methods [13–16,18,22–24]. We considered minimum soft constraint violation (SCV) that is mentioned by some state of art methods on ITC 2007 and ITC 2002 results. The results are calculated on 15 runs of the algorithm on every data instance.

Table 4. Comparison of HH-PSO with state of art methodologies on soft constraints violation of ITC 2002 data set

ITC2002	[22]	[23]	[24]	**HH-PSO**
1	831	86	79	**77**
2	577	**59**	73	250
3	662	116	**77**	100
4	975	135	175	130
5	792	**196**	292	326
6	730	**11**	133	359
7	698	**12**	170	574
8	610	**36**	82	383
9	549	**46**	69	256
10	645	85	83	**73**
11	953	76	81	**72**
12	579	134	118	**108**
13	826	120	**103**	257
14	796	**40**	253	377
15	700	**25**	123	369
16	684	**33**	64	359
17	856	249	170	**148**
18	624	**57**	61	288
19	758	**104**	186	383
20	697	**1**	94	33

Table 5. Comparison of HH-PSO with state of art methodologies on soft constraints violation of ITC 2007 data set

ITC 2007	[13]		[14]		[15]		[18]		[16]		**HH-PSO**	
	HCV	**SCV**	**HCV**	**SCV**	**HCV**	**SCV**	**HCV**	**SCV**	**HCV**	**SCV**	**HCV**	**SCV**
1	0	571	0	61	0	1482	0	**15**	0	1861	0	70
2	0	993	0	547	0	1635	0	0	39	2174	0	**50**
3	0	164	0	382	0	288	0	391	0	272	0	**150**
4	0	310	0	529	0	385	0	239	0	425	0	**228**
5	0	5	0	5	0	559	0	34	0	8	0	**4**
6	0	**0**	0	**0**	0	851	0	87	0	28	0	**0**
7	0	6	0	**0**	0	10	0	**0**	0	13	0	**0**
8	0	**0**	0	**0**	0	**0**	0	4	0	6	0	**0**
9	0	**1560**	0	**0**	0	1947	0	**0**	162	2733	0	1794
10	0	**0**	0	**0**	0	1741	0	**0**	161	2697	0	160
11	0	2163	0	548	0	240	0	547	0	263	0	**230**
12	0	178	0	869	0	475	0	**32**	0	804	0	34
13	0	146	0	**0**	0	675	0	166	0	285	0	302
14	0	0	0	0	0	864	0	0	0	110	0	390
15	0	1	0	379	0	**0**	0	**0**	0	5	0	690
16	0	**0**	0	191	0	1	0	41	0	132	0	642
17	0	2	0	**1**	0	5	0	68	0	72	0	10
18	0	**0**	0	**0**	0	3	0	26	0	70	0	70
19	0	1824	267	1862	0	1868	0	**22**	197	2268	0	1571
20	0	445	0	1215	0	596	655	2735	0	878	0	1295
21	0	**0**	0	**0**	0	602	0	33	0	40	0	302
22	0	**29**	0	0	0	1364	0	0	0	889	0	400
23	0	238	0	430	0	688	11	1275	0	436	0	**234**
24	0	**21**	0	720	0	822	0	30	0	372	0	562

It is observed that our proposed methodology is capable to produce less SCV than already reported SCV's on some data sets of ITC 2007 and ITC 2002. It is experienced that the overall soft constraint violation of HH PSO is between the highest and lowest reported SCV. It is experienced that our proposed method has a distance of feasibility zero and SCV is less and between the min-max limit of already reported SCV. Table 4 compares SCV of HH PSO with hyper-heuristic techniques for ITC 2002. Table 5 compares the hard constraint violation and soft constraint violation of HH PSO with techniques that participated in the ITC 2007 competition. The proposed method produced lesser SCV on 9 datasets of ITC 2007.

Conclusion and Future Work:

This research proposes a hyper-heuristic-based particle swarm optimizer for UCTP. Hyper heuristics are high-level methods that may select or generate or select low-level heuristics (LLH) for problem-solving. In this research, PSO is used as a higher-level methodology. Seven LLH are used. While one new LLH is proposed. We name it "least possible rooms left". It is concluded that HH PSO can generate a feasible solution for all data set instances of ITC 2002 and ITC 2007. HH PSO has the capacity to generate a feasible solution at the first stage that helps the algorithm to escape local minimums. When evaluating the performance of individual low-level heuristics, it is concluded that "least possible rooms left" have the potential to schedule more events than other LLH's. Therefore, we have used "least possible rooms" and "least saturated degree" in the first phase of solution construction to decrease the complexity of the dataset. It is concluded that this approach helps to construct feasible solutions. It is concluded that the increasing number of features required by events may increase the complexity of the dataset. The datasets that have a smaller number of rooms with a greater number of features are hard to solve. It is also concluded that datasets with a lesser number of events, but a greater number of students have greater conflict density. That may make the datasets hard to solve.

In this research, we have focused more on initial feasible solutions. and used only one strategy to improve the initial solution. In the future, we can explore more methods to improve the constructed feasible solutions. This may help us to reduce more soft violations. In this research, we have examined sequences of LLH and ignored the performance of individual Heuristic in the list and subsequences of the sequence. We can consider their impact in the future. We have worked on a specific problem model that contains the basic features of UCTP that are needed by many universities. It is debated during the research process to write a more general representation of hard and soft constraints that may fulfill as many requirements of most universities as possible. One more possible future direction could be to propose a more generalized representation of the university course timetabling problem.

Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for re-publication attached with the manuscript.
- The author has signed an animal welfare statement.
- Ethical Clearance: The project was approved by the local ethical committee in Universiti Sains Malaysia.

Authors' contributions:

Zahid Iqbal and Rafia Ilyas designed and directed the project. Rafia Ilyas performed the experiments. Chan Huah Yang revised the paper. Zahid Iqbal, Naveed Ahmed and Rafia Ilyas wrote the paper with input from all authors. Zahid Iqbal did the proofreading. All authors contributed the final manuscript.

References:

1. Hosny MI. A Heuristic Algorithm for Solving the Faculty Assignment Problem. 2013;
2. Schaerf A. Survey of automated timetabling. Artif Intell Rev [Internet]. 1999 [cited 2021 Feb 9];13(2):87–127. Available from: https://link.springer.com/article/10.1023/A:1006576209967
3. Burke EK, McCollum B, Meisels A, Petrovic S, Qu R. A graph-based hyper-heuristic for educational timetabling problems. Eur J Oper Res. 2007 Jan 1;176(1):177–92.
4. Burke E, MacCloumn B, Meisels A, Petrovic S, Qu R. A Graph-Based Hyper Heuristic for Timetabling Problems. Eur J Oper Res [Internet]. 2010;176:177–92. Available from: http://eprints.nottingham.ac.uk/371/
5. Burke EK, Elliman DG, Weare R. A university timetabling system based on graph colouring and constraint manipulation. J Res Comput Educ [Internet]. 1994 [cited 2021 Feb 9];27(1):1–18. Available from: https://www.tandfonline.com/doi/abs/10.1080/08886504.1994.10782112
6. Iqbal Z, Shahzad W, Faiza M. A diverse clustering particle swarm optimizer for dynamic environment: To locate and track multiple optima. In: Proceedings of the 2015 10th IEEE Conference on Industrial

Electronics and Applications, ICIEA 2015 [Internet]. Institute of Electrical and Electronics Engineers Inc.; 2015 [cited 2021 Feb 9]. p. 1755–60. Available from: http://ieeexplore.ieee.org/document/7334395/

7. Shiau DF. A hybrid particle swarm optimization for a university course scheduling problem with flexible preferences. Expert Syst Appl. 2011 Jan 1;38(1):235–48.
8. Thepphakorn T, Pongcharoen P. Performance improvement strategies on Cuckoo Search algorithms for solving the university course timetabling problem. Expert Syst Appl. 2020 Dec 15;161:113732.
9. Tan JS, Goh SL, Sura S, Kendall G, Sabar NR. Hybrid particle swarm optimization with particle elimination for the high school timetabling problem. Evol Intell [Internet]. 2020 Aug 28 [cited 2021 Oct 17];1:1–16. Available from: https://link.springer.com/article/10.1007/s12065-020-00473-x
10. McCollum B, Schaerf A, Paechter B, McMullan P, Lewis R, Parkes AJ, et al. Setting the research agenda in automated timetabling: The second international timetabling competition. INFORMS J Comput [Internet]. 2010 May 19 [cited 2021 Oct 17];22(1):120–30. Available from: https://pubsonline.informs.org/doi/abs/10.1287/ijoc.1090.0320
11. Goh SL, Kendall G, Sabar NR. Improved local search approaches to solve the post enrolment course timetabling problem. Eur J Oper Res. 2017 Aug 16;261(1):17–29.
12. Abdul Rahman S. Search Methodologies for Examination Timetabling. School of Computer Science and Information Technology. University of Nottingham. 2012.
13. Cambazard H, Hebrard E, Sullivan BO. Submission to ICT : Track 2. International Timetabling Compertition. 2007.
14. Mitsunori A, Koji N, Toshihide I. An Approach using General CSP Solver. ITC-2007 Track2. 2007.
15. Chiarandini M, Fawcett C, Hoos HH. A modular multiphase heuristic solver for post enrolment course timetabling. In: 7th International Conference on the Practice and Theory of Automated Timetabling, PATAT 2008. 2008.
16. Müller T. ITC2007 solver description: A hybrid approach. Ann Oper Res [Internet]. 2009 Oct 18 [cited 2021 Oct 17];172(1):429–46. Available from: https://link.springer.com/article/10.1007/s10479-009-0644-y
17. Goh SL, Kendall G, Sabar NR, Abdullah S. An effective hybrid local search approach for the post enrolment course timetabling problem. OPSEARCH [Internet]. 2020 Jun 20 [cited 2021 Oct 17];57(4):1131–63. Available from: https://link.springer.com/article/10.1007/s12597-020-00444-x
18. Nothegger C, Mayer A, Chwatal A, Raidl GR. Solving the post enrolment course timetabling problem by ant colony optimization. Ann Oper Res [Internet]. 2012 Feb 9 [cited 2021 Oct 17];194(1):325–39. Available from: https://link.springer.com/article/10.1007/s10479-012-1078-5
19. Ross P, Marìn-Blázquez JG. Constructive hyper-heuristics in class timetabling. In: 2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005 Proceedings. 2005. p. 1493–500.
20. Lissovoi A, Oliveto PS, Warwicker JA. On the time complexity of algorithm selection hyper-heuristics for multimodal optimisation. In: 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019 [Internet]. AAAI Press; 2019 [cited 2021 Oct 18]. p. 2322–9. Available from: https://ojs.aaai.org/index.php/AAAI/article/view/4071
21. Rossi-doria O, Paechter B. An hyperheuristic approach to course timetabling problem using an evolutionary algorithm. Evol Comput [Internet]. 2003 [cited 2021 Feb 9];(January 2014). Available from: https://www.researchgate.net/publication/228724044
22. Ross P, Marín-Blázquez JG, Hart E. Hyper-heuristics applied to class and exam timetabling problems. In: Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004. 2004. p. 1691–8.
23. Bai R, Blazewicz J, Burke EK, Kendall G, McCollum B. A simulated annealing hyper-heuristic methodology for flexible decision support. 4OR [Internet]. 2012 Mar 23 [cited 2021 Feb 9];10(1):43–66. Available from: https://link.springer.com/article/10.1007/s10288-011-0182-8
24. Rattadilok P. An investigation and extension of a hyper-heuristic framework. Inform. 2010;34(4):523–34.
25. Soria-Alcaraz JA, Ochoa G, Swan J, Carpio M, Puga H, Burke EK. Effective learning hyper-heuristics for the course timetabling problem. Eur J Oper Res. 2014 Oct 1;238(1):77–86.
26. Jonasson J, Norgren E. Investigating a Genetic Algorithm - Simulated Annealing Hybrid Applied to University Course Timetabling Problem. DEGREE Proj Technol [Internet]. 2016 [cited 2021 Feb 9];37. Available from: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-186364
27. Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks [Internet]. IEEE; [cited 2021 Feb 9]. p. 1942–8. Available from: http://ieeexplore.ieee.org/document/488968/
28. Irene SFH, Deris S, Mohd Hashim SZ. A combination of PSO and local search in university course timetabling problem. In: Proceedings - 2009 International Conference on Computer Engineering and Technology, ICCET 2009. 2009. p. 492–5.
29. Kanoh H, Chen S, H. Kanoh & SC. Particle swarm optimization with transition probability for timetabling problems. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in

Bioinformatics) [Internet]. Springer, Berlin, Heidelberg; 2013 [cited 2021 Oct 17]. p. 256–65. Available from: https://link.springer.com/chapter/10.1007/978-3-642-37213-1_27
30. Adrianto D. Comparison Using Particle Swarm Optimization and Genetic Algorithm for Timetable Scheduling. 2014;
31. Et al. M. An Analysis on the Applicability of Meta-Heuristic Searching Techniques for Automated Test Data Generation in Automatic Programming Assessment. Baghdad Sci J. 2019 Jun 20;16(2(SI)):0515.
32. Ilyas R, Iqbal Z. Study of hybrid approaches used for university course timetable problem (UCTP). In: Proceedings of the 2015 10th IEEE Conference on Industrial Electronics and Applications, ICIEA 2015. Institute of Electrical and Electronics Engineers Inc.; 2015. p. 696–701.
33. Song T, Liu S, Tang X, Peng X, Chen M. An iterated local search algorithm for the University Course Timetabling Problem. Appl Soft Comput J. 2018 Jul 1;68:597–608.
34. Yasari P, Ranjbar M, Jamili N, Shaelaie MH. A two-stage stochastic programming approach for a multi-objective course timetabling problem with courses cancelation risk. Comput Ind Eng. 2019 Apr 1;130:650–60.
35. Goh SL, Kendall G, Sabar NR. Monte carlo tree search in finding feasible solutions for course timetabling problem. Int J Adv Sci Eng Inf Technol. 2019;9(6):1936–43.
36. Pillay N. A review of hyper-heuristics for educational timetabling. Ann Oper Res. 2016;

الحل الفعال للجدول الزمني للمحاضرات الجامعية باستخدام محسن سرب الجسيمات استناداً على منهجية الإرشاد العالي

زاهد إقبال[*1,2] رافيا إلياس[2] هوا يونغ تشان[1] نافيد أحمد[2]

[1]كلية علوم الحاسوب ، يونيفرسيتي سينز ماليزيا ، 11800 ، بولاو بينانج ، ماليزيا.
[2]قسم علوم الحاسوب ، جامعة جوجرات ، غوجرات ، باكستان.

الخلاصة:
عادة ما تكون مشكلة الجدول الزمني للمحاضرات الجامعية (UCTP) هي مشكلة تحسين الإندماجية. يستغرق الأمر جهود يدوية لعدة أيام للوصول إلى جدول زمني مفيد ، ولا تزال النتائج غير جيدة بما يكفي. تُستخدم طرق مختلفة من (الإرشاد أو الإرشاد المساعد) لحل UCTP بشكل مناسب. لكن هذه الأساليب عادةً ما تعطي حلول محدودة. يعالج إطار العمل الاسترشادي العالي هذه المشكلة المعقدة بشكل مناسب. يقترح هذا البحث استخدام محسن سرب الجسيمات استنادا على منهجية الإرشاد العالي(HH PSO) لمعالجة مشكلة الجدول الزمني للمحاضرات الجامعية (UCTP) . محسن سرب الجسيمات PSO يستخدام كطريقة ذات مستوى عالي لتحديد تسلسل الاستدلال ذي المستوى المنخفض (LLH) والذي من ناحية أخرى يستطيع توليد الحل الأمثل. لنهج المقترح يقسم الحل إلى مرحلتين (المرحلة الأولية ومرحلة التحسين). قمنا بتطوير LLHجديد يسمى "أقل عدد ممكن من الغرف المتبقية" لجدولة الأحداث. يتم استخدام مجموعتي بيانات مسابقة الجدول الزمني الدولية (ITC) 2002 و ITC 2007 لتقييم الطريقة المقترحة. تشير النتائج الأولية إلى أن الإرشاد منخفض المستوى المقترح يساعد في جدولة الأحداث في المرحلة الأولية. بالمقارنة مع LLH الأخرى ، الطريقة LLH المقترحة جدولت المزيد من الأحداث لـ 14 و 15 من حالات البيانات من 24 و 20 حالة بيانات من ITC 2002 و ITC 2007 ، على التوالي. تظهر الدراسة التجريبية أن HH PSO تحصل على معدل خرق أقل للقيود في سبع وستة حالات بيانات من ITC 2007 و ITC 2002 ، على التوالي. واستنتج هذا البحث أن LLH المقترحة يمكن أن تحصل على حل معقول وملائم إذا تم تحديد الأولويات.

الكلمات المفتاحية: الجدول الزمني التلقائي ، الإرشاد العالي ، مُحسِّن سرب الجسيمات