Study of Hybrid Approaches used for University Course Timetable Problem (UCTP)

Rafia Ilyas, Zahid Iqbal

Abstract—**University course timetabling problem (UCTP) is important issue forever. Scheduling problems are considered and proved NP-Complete by different researchers. It is due to continuous changing of constraints to be fulfilled on rapidly changing data. Different hybrid state of art techniques and their use for university course timetable problem is investigated in this study. This paper also analyze occurrence of constraints and there ratio of similarity in recent research trend on university course timetabling problem. Constraints highly vary from department to department. This level of variance of constraints makes timetabling problem difficult to solve and NP-complete. In recent years, concept of hybridization of different methods increased. This study analyzed use of hybrid heuristic and meta-heuristic methods for university course timetabling problem. In this study, we categorize this hybridization into two main categories: local search hybridization with local search based approaches and population based hybridization with local search based approaches. It is observed that Population based methods (Genetic Algorithm (GA), Particle Swarm optimization (PSO) and Artificial Bee colony (ABC) etc.) are preferred in combination with local search (LS) based methods for university timetabling problem. Concept of hybridization of population based methods with local and other population based methods is adopted to eliminate demerits of both methods. Although, hybrid methods are difficult and also require more computational cost. Still hybridization is useful for finding better solutions.**

Index Terms — Scheduling, Timetable Problem, Optimization, Meta-Heuristics

I. Introduction

From last 65 years, auto timetable generation is most discussed issue. Still no well-known solution is found. It takes number of days to generate a good timetable manually. Timetabling problem rises in different areas e.g. exams, lecture/course, transportation, events and jobs etc. Among all timetabling issues, educational timetabling is most complicated task and widely discussed by researchers. It can be categorized into three classes (university/course timetable, school timetable and exam timetable). This paper focuses on university course timetable problem (UCTP).

UCTP can be defined as: allocation of finite set of room, courses, students and lecturers to finite set of timeslots that must fulfill some *constraints* [1-8]. Constraints are mainly divided into two categories: *soft constraint* and *hard constraints [5-6]*. Hard constraints are those that should must be fulfilled, without fulfilling hard constraints timetable cannot be accepted. Soft constraints have lower priority then hard constraints. Timetable can be accepted without satisfying soft constraint. Requirement for UCTP varies from area to area, university to university and even from department to department. Due to which generic solution of UCTP is difficult to find. Fulfillment of constraint makes UCTP more difficult. This study will analyze use of constraints in context to solve UCTP intelligently. From the progress in intelligent techniques, research is diverted to use intelligent methods in different domains for optimized results [9-12]. Recent research demonstrated remarkable performance of intelligent methods in solving problems, where classical methods may fail or produce infeasible solution. Intelligent methods i.e. Evolutionary Algorithms, Swarm Intelligence, Artificial Neural Networks produced optimized results in different domains [9-12]. Similarly, state of art intelligent techniques are used to solve UCTP [1] [3] [5-7]. Some state of art techniques are Heuristics and Meta Heuristics. This paper will also study use of hybridization of these techniques used for solving UCTP. Section II will hold detail of variations in UCTP. Section III review latest exiting techniques used for UCTP. Section IV presents analysis of literature studied in III. Section V holds conclusion and future work.

II. Variations of problems

UCTP is about allocating lecturer to finite resources. Every university has its own requirement set for resource allocation. Some hard constraints that are fulfilled by every technique are:

- No student can take more than one class at same time.
- There will be one class in a room at particular timeslot.
- No teacher can take more than one class at same time slot.

Every proposed technique for UCTP, fulfill some of particular constraints. This study covers variations of UCTP. Highly considered variations are:

A. Grouping and sectioning: In universities, one course may be taught multiple times in a semester. In this case lectures are assigned to sections instead of single student. Section problem is handled in [3] [5].

B. Post-Pre assignment of courses: Some universities firstly took admissions, then assign course (post-assignment).While other assign course first, then took admission (pre-assignment). In both cases, timetable management will be different. Both scenarios are solved in [4].

C. Fixing rooms in fixed time slots: For special lectures, time slots and rooms are fixed. In that time, no other class can be taken. This problem is tackled in [5].

D. Student's Preferences: Some universities take care of student's preferences. That one student may or may not take two consecutive classes; students may or may not have only one class in a day etc. [5-7]

E. Teacher's preferences: Some universities take care of teacher preferences. It includes availability of a teacher in a given time slot. This type of UCTP is solved in [5].

It is observed that requirement set to solve UCTP changes from university to university and even from department to

978-1-4799-8389-6/15/$31.00 ©2015 IEEE

department. This level of variance of constraints turns each UCTP into a distinct problem. One can say that every research focused on distinct UCTP and proved performance of practiced method for that particular problem set and dataset only. Literature review analyze hybrid different techniques for UCTP.

III. Solution Approaches

Different solution approaches are used to solve different UCTP. This paper will study recent solution approaches used to solve UCTP that can be mainly categorized into heuristic and meta-heuristic based approaches.

3.1 Heuristic: Where traditional methods fail to give solution, heuristic is applied there to find out near optimal solution. It is based on random search mechanism. Although, heuristic does not promises globally best solution, still it is beneficial when traditional methods fail. There are typically two heuristic approaches named: *constructive heuristic* and *improvement heuristic*. Constructive heuristic take start from empty solution and then construct solution in incremental way to create a complete solution. While improvement heuristic starts from random solution and then improve it in iterative manner. Mostly, both methods are used together [13].

3.1.1 Model Based Heuristics: A new model, based on classic heuristic approach to solve UCTP is proposed in [4]. Algorithm controls constraints and resources for scheduling (one-by-one). Algorithm solves timetable by selecting particular course causing clash. This type of solving method is known as Constructive Heuristics. And after removing necessary clashes algorithm also fixes a course to a particular time slot where it is required (reserved slot). This Approach used constructive heuristic at first step and then improvement heuristic on second step. Timetable is generated by in incremental manner. Progress of resource adjustment process is checked, if conflicts are not removed it gives message that system is enable to remove conflicts. Fig.1 shows working mechanism of this algorithm. System developed by using this algorithm is checked on real world data by deploying it in Al-Faisal University, Saudi Arabia. This algorithm may not remove all clashes. In most scenarios, heuristic is used in combination of other methods. It is because of local heuristics draw-back is to stick to local optima and not promising global optimal solutions. Procedure of integrating heuristics in other methods is called Meta-Heuristic.

3.1.2 Meta-Heuristic: These techniques are usually used to solve hard problems. Recently, meta-heuristic techniques are most often used to solve UCTP. *Meta Heuristic is an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information for finding near-optimal solutions.* (Osman 1996). These techniques are divided into two classes [13]: *Local search based* methods deals with single object in one iteration. Local search based techniques are also called single solution based techniques. In this approach, single solution is created, then modified using local search. *Simulated Annealing (SA), Tabu Search (TS), Multi-Neighborhood, Local Search (LS)* etc. are local search based methods.

Population based search deal with population of objects on each iteration. Multiple Solutions are built to explore search space and move toward optimal solution. *Genetic algorithm (GA), Artificial Bee Colony (ABC), Particle Swarm optimization (PSO)* are population based approaches.

Now a day, hybridization is highly adopted. For UCTP, different meta-heuristic methods are used in combination of some other meta-heuristic methods. This paper mainly divides hybridization methods used for UCTP into three parts: Local Search with Local Search methods; Population Based Search with Local Search; and Population Based Search with population based search.

3.1.2.1 Local search based methods with Local search based methods: Many local search based method are hybridized with other local search based methods for UCTP. Some recent combinations are SA with neighborhood search [8], SA with Round Ribbon and TS with local search etc. Below is description of Local Search methods and their hybridization.

A. Simulated Annealing (SA): is inspired by heat checking and cooling of materials method to step up crystal size [14]. SA is an optimization method that can be applied to arbitrary search and problem spaces. Like hill climbing algorithm, SA also works with single initial individual. SA is also used for UCTP. Slow change of cooling factor makes SA search process slow and SA may stop search before optimal solution. That's why SA is used in combination with other methods [8] [15].

i. Simulated Annealing with multi-neighborhood: Multi-Neighborhood is used to eliminate demerits of SA to solve UCTP [8]. SA is based on moves, used in search space to find appropriate space for courses and neighbors. Lectures can have two moves (P-move, R-move): P-move changes time and R-move changes rooms of lecture. Lectures are selected and swapped randomly with any of its neighborhood only.

Multi-neighborhood search is used to cover slow cooling effect of SA. Searching two types of moves in random neighborhood makes this approach efficient. Performance of SA with multi-neighborhood search is compared with SA with single neighborhood search. It is observed that single search neighborhood "P" performs better than "R" and SA with multi-neighborhood performed better than SA with single neighborhood, still ratio of violating soft constraints of SA with multi-neighborhood is inadequate.

ii. Dual-sequence Simulated Annealing (DSA) with Round Robin (RR): SA is used with RR algorithm for solving UCTP [15]. DSA-RR has two phase heuristic: constructive and improvement. SA starts from empty solution, generated by using *Least Saturated Degree (LSD) heuristic* [15]. LSD first considers those lectures that have least degree of remaining timeslots left. Three neighborhood structures are traced for improvement of constructed solution. RR method is used to improve neighborhood selection structure for optimal solution.

It is observed that DSA-RR outer perform than extended great deluge, iterative improvement and genetic algorithms with local search etc. Performance is analyzed on "*Socha*" dataset. It is observed that system generate different timetable on every run. This is because of its random selection.

B. Tabu Search (TS): works like SA. Difference is that TS maintains a Tabu list of all previously visited solutions. It avoids re-finding of those solutions. This method enforces to search new solution even if their quality is poor. It avoids premature convergence at local optimal. TS is integrated with different local search methods for UCTP.

i. *Tabu search with iterative local search (ILS):* Adaptive TS is used with ILS for solving UCTP. Constructive and improvement both approaches are used in [16]. Firstly, a solution is created using fast greedy search. TS used adaptive method to reduce conflicts ratio and after then for finding optimal solution neighborhood are searched using ILS for optimal solution. ILS use swapping mechanism to find optimal solution. ILS is mainly used to find best non-tabu solution. Results show that hybrid TS performed better than TS and ILS alone. Methods are evaluated on ITC dataset.

3.1.2.2 Local Search with Population Based Search: Recently, trend to use population based search strategies is increased. Population based methods are used with LS to removes flaws of both methods. Population based search methods are inspired by nature. Different population based methods and their integrations for UCTP are discussed below.

A. Population based local search (PB-LS)" In 2013, local based search and population based search are integrated for UCTP [6]. Integration is used to eliminate flaws of both techniques. Process of solving UCTP has two parts (initialization, exploring solution). Constructive heuristics based on three steps. *(LSD, neighborhood search, TS)* is used to create initial solution. At step one, classes having conflicts are sorted in order to frequency of students with unscheduled classes. Then highest frequency conflict is selected to search for feasible solution. If feasible space is not found, LSD select time slots randomly. If solution if found in first step other two will be skipped. Otherwise hill climbing algorithm is used to find solution. Neighborhood solution is generated by swapping time slots. If swapping remove conflict and give better solution without disturbing hard constraints and quality of timetable. Solution will be accepted. This step can be repeated only 10 times if solution not found third step will run. Otherwise third step will be ignored. In step 3, TS searches like step 2 but also maintain Tabu list. If solution found on first step other two steps will be ignored and if solution is found on step 2, step 3 will be ignore. PB-LS Explores iteratively. PS-LS use 8 neighborhood structures to find feasible solution (briefly discussed in [6]). Fig. 2 shows model of PB-LS. Results of PB-LS are compared with previously used algorithm named gravitational emulation local search (GELS). It is observed from results that PB-LS works better then GELS. PB-LS worked well on small data set but performance may be affected with large data set. *Socha* dataset is used to evaluate performance of above model.

B. Genetic algorithms (GA): are used to find optimal solution. GA uses natural operations i-e mutation, cross-over for finding optimal solution. GA was introduced in 1975. Some basic terms used in Genetic algorithms are: *Chromosome:* An individual piece of information, present in search space to solve problem; *Population:* Set of chromosomes; *Fitness function:* check fitness of each individual chromosome fitness function is used and *Cross Over and Mutation*: used to generate new population from fittest parents. In GA firstly, population is initialized. Most often, initialization is performed randomly [5]. Then fitness of each individual is calculated using fitness function. Individual with strong fitness are selected for new generation. Cross over and mutation is applied to generate new population from fittest parents. Recent research shows that GA are used with other methods to get promising results in different domains [9]. GA is also used widely with different techniques (Greedy search, SA and Case-Based reasoning (CBR) etc.) to solve UCTP [5] [17-19]. Here is some recent hybridization of GA.

i. *GA with greedy initialization:* informed genetic algorithm (IGA) is proposed by [5] to solve UCTP. Greedy algorithm and directed mutation is used for more effective results. IGA have two stages. In stage one, greedy algorithm is used to build initialized timetable. All clashes excluding student constraints are handled in stage one. Greedy technique selects lectures on the basis of given timeslot `ratio. If it fails to find solution, method will be repeated for specific trail period. Fig.4 shows model of IGA. Directed mutation perform following tasks

- Produce N offspring from N initialized individuals.
- Mutate gene randomly to reduce conflicts with some other time slots that do not violate any hard constraint.
- Moves genes to remove collective conflicts genes with collective conflicts are swapped with each other.
- Swap genes with no clash to increase level of diversity.

Second stage works on stage one output data. At stage two, all clashes are removed including student constraints. Again directed mutation is performed but additional swapping of gene with no violation is not performed on this level. Algorithm will keep working until best solution is not found. As a result of directed mutation and additional swapping student conflicts decreased by 8.40%. It is observed that IGA performs better than GA. But IGA may keep on running or stops after a specific trail period with no best far solution.

ii. *Genetic Algorithm with Case-based reasoning (CBR)*: In 2005, CBR with GA is proposed in [17] for UCTP. Case base stores previous timetable to deal with future problem timetabling. At first step, on problem data algorithm tries to find solution from previously stored solution. If solution found then algorithm terminates, if not then it selects some near solution timetable from database and apply GA to modify it according to given conditions. When new solution is found through GA, human assistance is needed to check whether new created solution is acceptable or not. If solution is acceptable, it is stored in case base for future use. Fig. 3shows procedure of CBR- GA. On one hand, this process reduces computational complexity. It is because if solution exists then it can be reused. And no need to find new solution. But on the other hand, CBR requires huge amount of memory to save all previous solutions.

C. *Swarm Intelligence (SI):* is integrated behavior of multiple agents in a particular scenario. Its concept was given in 1989 [20]. There is no clear definition of SI. Different SI algorithms are PSO, ABC and Fish Swarm Intelligence Algorithm etc. Motivation to create such algorithms is taken from nature. Now a day, these algorithms are used for solving different problems, including UCTP to get optimized solution.

i. Particle Swarm Optimization (PSO) with Local search (LS): In 1995, development of Particle swarm optimizer (PSO) was done by Kennedy and Eberhart. [21].An individual bird in crowd act as a particle and swarm is group of particles. PSO works on food searching strategy used by birds. Particles move

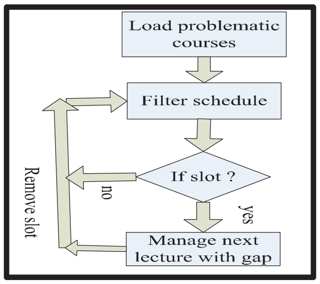

Fig. 1 Model based heuristic

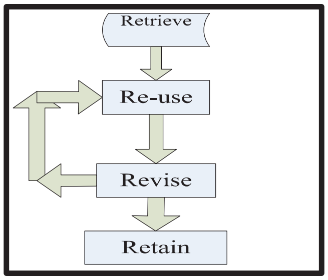

Fig. 3 Model CBR-GA

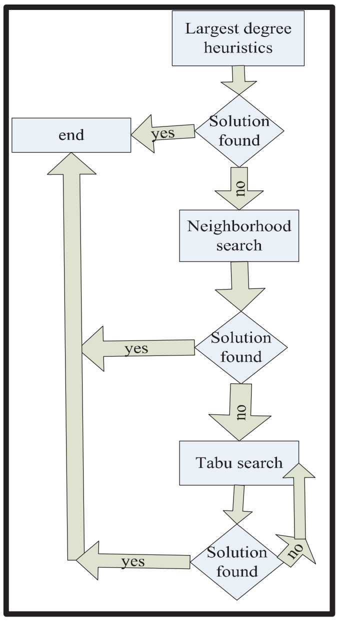

Fig.2 PB-LS Model

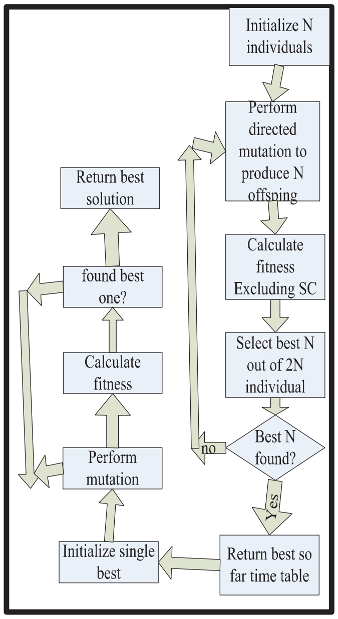

Fig.4 IGA Model

into search space in different dimensions. Every bird saves two basic set of information (global best, personal best). Global and personal best are retrieved on the basis of fitness, calculated by objective function [21-22]. Research trends shown success of optimization methods in many areas. Recenlty, intresting research has been performed to get optimal results using PSO [10]. PSO with some modification is used also for UCTP. In UCTP, every individual lecture with some constraint is used as particles. These particles move in search space to fulfill given constraints and find out optimal solution. For this purpose, every time when a particle changes its position its fitness value is calculated. Comparison is made between each particles own previous best and also with globally best solution. Fitness of every particle is calculated on the basis of personal and global best. This fitness is used to move the particle toward optimal solution in search space. Original PSO, Standard PSO (SPSO), PSO using LS and SPSO with LS is used in [22] for UCTP by fulfilling additional powerful hard constraints e.g. class reservation, student and teacher preferences etc. It is observed that SPSO outer performs than PSO alone. PSO stuck with larger data sets and do not improve fitness. But PSO with LS (PSOLS) and SPSO with LS (SPSOLS) perform better than PSO and SPSO [22]. Applying LS with Population based methods improve results. [7] [22]

ii. Modified Fish Swarm Intelligent: a modified fish swarm intelligence system is used for UCTP by [7]. Fish swarm algorithm is inspired by fish search for food on the basis of its optic space of a fish. Where optic space is divided into three categories (more populated, empty, and less populated). Behavior of solution (fish) varies for every category. There are three categories of fish behavior w. r. t to its optic space i.e. swarming, chasing and searching. When search space is empty chasing will be performed using steepest descent algorithm otherwise swarming will be applied using great deluge algorithm. In populated search space, Searching is performed in three directions (Contraction-External, Reflection and Expansion) using Nelder Simplex Algorithm. Nelder-Mead Simplex Algorithm is again used to improve quality of solutions. Improvement in search solution is 2% to 40%.

iii.Modified Artificial Bee Colony (ABC): is inspired from bee's behavior in selection of food [23]. Bee colony works in three different groups of bees (onlooker, employed and scout). Onlookers analyze behavior of employed bees that search food linked with some particular food source. Scout randomly searches new food sources. Optimal food source is solution of the problem. Each employed bee is associated with a solution the problem. [23] A modified form of ABC is used to solve UTCP [24]. Multi-swap concept proposed by [25] is used at onlooker level to improve results of ABC. It is observed that modified-ABC gives better solution than previously used methods, but requires more memory, as it also uses onlookers to find optimal solution is search space and perform multi swapping on search space where better fitness is found [24]. From above literature, we observe many similarities and dissimilarities of timetabling problem. Section IV will exhibit analysis of literature.

IV. Analysis

This section will demonstrate constraints and their level of similarity and techniques and use of these techniques.

A. Analysis of Constraints: As discussed above that timetabling problems vary from each other on the basis of constraints fulfilled in generation of timetable. Here, level of similarity of constraints fulfilled by different methods is examined i.e. to which extend same constraints occur in UTCP. Table 1 holds constraints and its fulfillment through different techniques. To analyze occurrence of constraints, 18 used constraints are taken. It is observed that only 4 of 18 are highly relevant used by approximately every researcher, other constraint occurs in 50% papers. This high level dissimilarity of constraints makes generic solution of timetabling problem hard. Due to highly variance of constraints this issue is called NP-complete. Table 2 shows data sets used to evaluate models for timetabling problem. It is observed that different instances of Socha dataset are used by four researchers, three researchers

Table 1
Techniques VS Constraints

Constraint no.	Papers / Constraints	Paper handling these constraints
C1	No more than one class at a same time	[4-8] [15-16] [19] [22][24]
C2	No more than one teacher class	[4-6][8][15-16][19] [22]
C3	Class should be taken in room capacity and features	[4-8][15-16] [19][22][24]
C4	Student should not have one class in a day	[6][16][24]
C5	No more than two consecutive classes	[6] [8] [15-16] [24]
C6	Specific classes can be arranged in specific time slot.	[4-5] [15] [22]
C7	Prioritize time slots. If lectures are not adjusted prioritize it to slot to adjust it.	[4]
C8	At last time slot, there should be no student class	[6][24]
C9	At last time slot, there should be student class.	[7]
C10	Departmental Schedules	[5][22]
C11	No duplication of course	[24]
C12	No double booking of rooms	[5-7] [15][16] [19][24]
C13	No of classes per week should must be scheduled	[5] [15] [19]
C14	Student should have one class in a day	[7]
C15	More than two consecutive classes	[7][15]
C16	Student preference	[5][22]
C17	Teachers preferences	[5-7][22]
C18	Room stability. Try to assign a room for particular lecture for whole semester	[16]

C1, C2, C3 and C12 are handled as hard constraints in most research. So, to get an acceptable solution fulfilling these constraints is necessary While other constraints are handled as soft constraints and penalty values is calculated for breaking up these constraint. Also these constraints vary from department to department. It can be observed that appearance of these constraints is highly varying for every referred research.

Table 2
Techniques VS datasets

Research	Institute/data sets	Reference
Model based heuristic	Al-Faysal university	[4]
Informed genetic algorithm	University data	[5]
Population based local search	Socha Benchmark	[6]
Fish Swarm Intelligence	ITC 2007	[7]
Multiple-neighborhood with simulated annealing	Experimental data	[8]
Dual Simulated Annealing with Round Robin	Socha Benchmark	[15]
Tabu Search with iterated local search	ITC 2007	[16]
GA with case based reasoning	University data	[17]
Guided genetic algorithm	-	[19]
Particles Swarm optimizer and its variation	Socha benchmark	[22]
Artificial Bee colony	Socha Benchmark	[24]

used real world university datasets, two researchers used ITC2007 dataset and one researcher used experimental data.

B. Analysis of use of methodologies: From section III, we divide use of different problem solving models for UCTP into three major categories. Researchers used different local and population based methods are used with other such approaches to removes flaws of existing techniques. Table 3 shows that four local search techniques hybridized with local search methods [4] [8] [15-16], named SA, TS are merged with multi-

Table 3
Local Search integration with Local search methods

Technique	Hybridized techniques	Reference
Constructive heuristic	Improvement heuristic	[4]
Simulated Annealing	Multi-Neighborhood Search	[8]
Simulated Annealing	Round Robin	[15]
Tabu Search	Iterative Local search	[16]

Table 4
Population Based Search integration with local search

Technique	Hybridized techniques	Reference
Genetic Algorithm	Greedy search	[5]
Population based	Multi-Neighborhood Search/Tabu Search/largest degree heuristic	[6]
Fish Swarm Optimization	Steepest descent, Nelder Mead Simplex, Great Deluge search	[7]
Genetic Algorithm	Case based reasoning	[17]
Genetic Algorithm	Probability based selection	[19]
Particle swarm optimization	Local search	[22]
Artificial Bee Colony	Multi-swap	[23]

Table 5
Names of Techniques VS Reason of hybridization

Research	Reason of hybridization	Reference
Model based heuristic	To obtain optimal solution improvement heuristic is used after constructing feasible solution.	[4]
Informed genetic algorithm	To construct solution fast. And to find optimal solution	[5]
Population based local search	To overcome intensification and exploitation	[6]
Fish Swarm Intelligence	To check diversity and convergence of solutions for finding optimal solution	[7]
Multiple-neighborhood with simulated annealing	To deal with slow cooling mechanism of simulated annealing. Multi-neighborhood search makes search procedure fast.	[8]
Dual Simulated Annealing with Round Robin	To improve neighborhood search and to find fast and optimal solution	[15]
Tabu Search with iterated local search	To improve neighbor search and to find fast optimal solution	[16]
GA with case based reasoning	To reduce computation cost by saving previous solution rather than re-creating them.	[17]
Guided genetic algorithm	To deal with random behavior of Mutation and to produce stable mutation	[19]
Particles Swarm optimizer and its variation	To produce optimal solution with exploration. Due to fast searching capability of local search	[22]
Artificial Bee colony	To find optimal solution	[24]

neighborhood search, and ILS respectively. And multiple neighborhood search structures are promoted to make search fast. Table 4 shows Remaining 7 research works, merged local based and population based methods Ratio of hybridizing local search method with population based methods is greater than merging local search with local search.

It is because local search stuck in local optima and may not be able to find out globally best solution. But on the other hand, population based search takes more memory and time to find optimal solution. Table 5 holds reason of hybridizing of considered methods. It is observed that local search methods make solution exploration faster. And population based methods are used to find globally optimal solution where local search methods stuck on local optima.

V. Conclusion

Different intelligent techniques used for university timetabling are analyzed in this paper. It is observed that from last few years concept of hybridization is increased. Very few researchers use an individual technique alone to solve timetabling issue. Different local and population based techniques are merged with each other to eliminated disadvantages of one another. Constraints are also analyzed in this study to observe the probability to work on generic solutions for university course timetabling problem. For this, we analyze 10 recent studies and their trends. There are 40% of similar constraints that are fulfilled as hard constraints by 95% papers. Other 60% constraints vary from university to university. This ratio shows the difficulty in finding generic solutions for UCTP. One of main problems of university course timetabling problem is that all research solutions are scenario based and can work well on a particular dataset only and may not be useful for some other institute/dataset. It is concluded that if problem set will be generalized than it will be more effective to find its solution.

In future, we will focus that to what extend problem set for UCTP can be generalized? And what is Possibility of better solutions by hybridizing population based methods with population based?

References

[1] Alberto Colorni et al, "a genetic algorithm to solve the timetable problem", computational optimization and applications journal

[2] Pertrovic s, burke e (2004) educational timetabling. In: handbook of scheduling: algorithms, models and performance analysis pp. 41-45

[3] Ricardo santiago-mozos , sancho salcedo-sanz_ ,mario deprado-cumplido and carlos bouso~no-calz_on , a two-phase heuristic evolutionary algorithm for personalizing course timetables: a case study in a spanish university (2005)

[4] Tahir afzal malik, sajjad sadiq. Hikmat ullah khan, dynamic time table generator conforming constraints: a novel approach, "iccit", springer, 2012

[5] suyanto, an informed genetic algorithm for university course and student timetabling problems, "icaisc 2010" part ii, pp. 229-236

[6] Anmar abuhamdah, masri ayob, graham kendall, population based local search for university course timetabling problems, applied intelligence (2014)

[7] Humza turabeih, salwani abdullah, barry mccollum and pual mcmullan, fish swarm intelligent algorithm for course timetabling problem, springer-verlag berlin Heidelberg Rskt 2010, pp. 588-595

[8] He yan, song-nian yu, a multiple neighborhoods-based simulated annealing algorithm for timetable problem, springer-verlag berlin heidelberg, gcc 2003, pp. 474-481 2004

[9] Zahid Iqbal et al(2013), Efficient Machine Learning Techniques for Stock Market Prediction Int. Journal of Engineering Research and Applications ISSN : 2248-9622, Vol. 3, Issue 6, Dec 2013, pp.855-867

[10] Zahid Iqbal et al(2015), "A diverse clustering Particle Swarm Optimizer for Dynamic Environment: to Locate and Track Multiple Optima"10th intr. Conference on industrial Electronics and Applications

[11] Zahid Iqbal et al(2015), "An Efficient indoor navigator technique to find optimal route for blinds using QR codes"10th intr. Conference on industrial Electronics and Application

[12] Zahid Iqbal et al, "A Systematic Mapping Study on OCR Techniques" Int. Journal of Computer Science and Network Security, ISSN: 2345-3397, Vol. 2, No. 1, Jan 2014.

[13] Syariza abdul rahman, search methodologies for examination timetabling, thesis submitted to the University of Nottingham for the degree PHD, 2012

[14] Kirkpatrick, s.; gelatt jr, c. D.; vecchi, m. P. (1983). "Optimization by simulated annealing". Science 220 (4598): 671–680.

[15] Salwani abdullah, khalid shaker, barry mccollum, and paul mcmullan, dual sequence simulated annealing with round-robin approach for university course timetabling, p. Cowling and p. Mer (eds.): evocop 2010, lncs 6022, pp. 1–10, springer-verlag berlin heidelberg 2010

[16] Zhipeng l¨u, jin-kao hao , adaptive tabu search for course timetabling, *european journal of operational research 200(1): 235-244, 2010.*

[17] Alica grech, julie main, a case-based reasoning approch used to formulating university timetables using genetic algorithms, springer-verlag berlin heidelberg 2005

[18] Shengxiang Yang, *Member, IEEE*, and Sadaf Naseem Jat (2011), Genetic Algorithms with guided and local search strategies for university course timetabling, "IEEE transaction on systems, man and cybernetics

[19] Sapru,v. ,time table scheduling using genetic Algorithms employing guided mutation, computational intelligence and computing research (iccic), 2010 IEEE international conference on 2010 , pp. 1 - 4

[20] Beni, g., wang, j. Swarm intelligence in cellular robotic systems, proceed. Nato advanced workshop on robots and biological systems, tuscany, italy, june 26–30 (1989)

[21] Kennedy, j, eberhart, r, particle swarm optimization, neural networks, 1995., IEEE international conference on (volume:4), pp. 1942 – 1948

[22] Ruey-maw chen, hsiaso fang sihh, solving tuning constriction particle swarm optimization, algorithms 2013 (www.mdpi.com/journal/algorithms)

[23] D. Karaboga, an idea based on honey bee swarm for numerical optimization, technical report-tr06,erciyes university, engineering faculty, computer engineering department 2005

[24] A.l.bolaji et al, a modified ABC algorithm for pre-ctp, springer-verlag berlin heidelberg 2013

[25] Al-betar, m. Khader, a.t. Muslih, o : a multiswap algorithm for university timetabling problem: in : 2012, computer & information science (iccis), 2012 international conference on (volume:1) ieee (2012)

[26] Socha K, Knowles J, Sampels M (2002) A max-min ant system for the university course timetabling problem. In: Ant algorithms. Springer, pp 1–13

[27] V Dino matijas et al, university course timetabling using ACO: a case study on laboratory exercise, springer 2010

[28] Global optimization algorithms– theory and application, thomas weise 2nd edition Version: 2009-06-26

[29] Cowling p, kendall g, soubeiga e (2000) a hyperheuristic approach for scheduling a sales summit. In: selected papers of the third international conference on the practice and theory of automated timetabling.

Zahid iqbal received the B.S. degree in Computer Science from University of Punjab and M.S. degrees in Computer Science from FAST University Pakistan in 2010 and 2012 respectively. He was a lecturer with Department of IT, University of Punjab, Punjab, Pakistan. From 2012 to till date, he is a Lecturer with the Faculty of Computing and IT, University of Gujrat, Punjab, PK. His current research interests include evolutionary algorithms, swarm intelligence, artificial neural networks, computational intelligence in dynamic and uncertain environments and real-world applications.

Rafia Ilyas received the MSc degree in Computer Science from University of Gujrat in 2013. She is doing MPhil in Computer Science from University of Gujrat. Her research interest focuses on artificial neural networks, computational intelligence, scheduling and timetabling.