First International Conference on Emerging Trends in Engineering, Management and Scineces"
December 28-30, 2014 (ICETEMS-2014) Peshawar, Pakistan

Two Phase Cascading (TPC) Technique For Solving Curriculum Based Course Timetabling Problem

Muhammad Hannan
City University of Science & IT, Peshawar, Pakistan
m.hannan@cusit.edu.pk, sir_hannan@yahoo.com

Muhammad saeed shehzad
City University of Science & IT, Peshawar, Pakistan
mss@cusit.edu.pk

Abstract

The Curriculum-Based Course Timetabling (CB-CTT) problem is a weekly based problem where set of courses are scheduled in lectures, and every lecture have room and timeslot. This schedule must satisfy all the hard constraints and minimize the soft constraint as much as possible. This research proposed a Two Phase Cascading (TPC) technique for solving CB-CTT problem. Phase I creates a course wise cluster with the help of hierarchical clustering technique and assigns priority to each cluster and courses in a cluster. Phase II generates the timetable by selecting the higher priority cluster and course recursively. We have tested our proposed TPC technique on different datasets of International Timetabling Competition (ITC-2007). Experimental results show that proposed TPC technique perform better in terms of time and constraints for simple as well as complex datasets as compared to existing technique.

Keywords

Clustering, Scheduling, Timetabling, Constraints, Course Timetabling Problem

1. Introduction

Timetabling problems are known to be one of the most critical real time problem and divided into different forms, e.g., nursing timetable, sports timetable, flight schedule timetable, transportation timetabling, educational institute timetabling (examination timetabling, courses timetabling), , …etc, by considering different resources, processes and constraints.

1.1 University Course Timetabling (UCT)

Over the last few years UCT problem take too much attention due to its importance in the educational setups where administration do more efforts and activities on regular basis. In NP-Hard category UCT is represented as combinatorial optimization problem. UCT takes more efforts in the form of humans and computing so the required solution is more efficient and effective (Qu and Burke 2009). Post enrollment and curriculum based timetabling problems are the two forms of UCT (Abdullah, Burke et al. 2007).

1.1.1 Post Enrolment Based Course Timetable (PEBCT)

In general, post enrollment based course timetable (PEBCT) problem is a complex combinatorial optimization problem. In PEBCT students are registered in various courses offered by its university or department. This timetable creation technique is generally favored by a lot of educational Institutions because it gives the choice to students to select their courses by it and make sure that all the existing resources are well organized and effective when creating the timetable.

1.1.2 Curriculum Based Course Timetabling Problem (CB-CTP)

The CB-CTP is a weekly based timetabling problem where set of courses are scheduled in lectures, and every lecture have room, timeslot, course which satisfied all the HC and SC should be reduced (Di Gaspero, McCollum et al. 2007). Educational institute offered courses to different curricula or batches so the conflicts occurs among different courses are the constraints of CB-CTP so it is not related to post enrollment problem (Jazzar 2012).

2. Literature Review

From the last forty years huge efforts, methods and techniques has been done to find the efficient and constraint free solution to solve timetabling problem, however this problem are yet still in the focus of research (Wahid and Hussin 2014). (MirHassani and Habibi 2013) defines some methods which are used to solve the University course timetabling problem are Population Based Approaches include Genetic Algorithm (GAs), Hormonic Search Algorithm, Memetic Algorithm and Ant Colony Optimization (ACO). Meta-Heuristic Method includes Simulated Annealing (SA), Tabu search algorithm (TS), Great Deluge (GD), Hybrid Meta-Heuristic, Variable neighbor search (VNS) and Local Search. Operation Research (OR) based techniques include Graph based approach, constraint based technique (CSPs) and Integer programming/Linear programming (IP/LP) method. Modern intelligent approaches includes Hyper-Heuristic, combinational approaches, fuzzy based, Knowledge based, Case-based reasoning, artificial intelligence (AI) approaches and etc (Obit, 2010).

(Müller 2009) describes a multi-phase local-search algorithm combining a constructive forward search to obtain a feasible solution with successive local-search steps based on Hill-Climbing (HC), Great Deluge (GD) and SA. (Bellio, Di Gaspero et al. 2012) propose a hybrid local-search algorithm which alternates SA with dynamic Tabu Search (TS) with shifting penalties. The work is supported by an extensive statistical analysis on the e_ect of parameters. (Lü and Hao 2010) propose a three-phase hybrid algorithm which improves a greedy feasible solution through alternate intensi_cation (TS) and perturbation. The trade-o_between intensi_cation and diversi_cation is controlled by two parameters that are adjusted based on the past performance. (Abdullah, Turabieh et al. 2012) describe a hybrid meta-heuristic based on GD and an electromagnetic-like mechanism (EM) that performed very well on various tracks of the ITC2007 competition.

3.0 Problem Description

The CB-CTT builds with the following entities (Kohshori and Abadeh 2012)

- **Days, Timeslots and Periods** – We have no of days in a week for any educational institute. Each day is partitioned into fixed set of timeslots. Each timeslot and day makes a unique period.
- **Courses** – Each course in the timetable has a number of lectures which is assigned to unique periods. A teacher is assigned to each course and students registered in different courses. The course is not assigned to a period which is unavailable.
- **Rooms** – We have a numbers of rooms with a specific capacity.
- **Curricula** – A group of Courses is assign to a Curriculum in such a way each pair of courses belongs to curriculum has same students. Lectures in the common curriculum cannot be taught together.

The final problem solution should be that all courses have a lecture within available day and timeslots. All courses must satisfied all the hard constraints and minimize the soft constraints. Some of the hard and soft constraints are (Lü and Hao 2010).

- **Lectures** – Lectures of All Courses should be scheduled in distinct room and time slot.
- **Room Occupancy** – In a room at the same timeslot two lectures of courses are not to be assigned.
- **Conflicts** – At the same time slot all teacher and students have only one lecture or event.
- **Availability** – Lecture or events must be schedule to that time slots in which teachers are available.
- **Room Capacity** – The capacity of classrooms not exceed to the strength of students in that course.
- **Room Stability** – all course lectures must be use the same room.
- **Minimum Working Days** – all lectures of course must satisfy the minimum working days.
- **Curriculum Compactness** – All lectures within the curriculum must be adjacent with each other.

4. Two Phase of Proposed Solution

The implementation of our approach algorithm is in c#.Net where we first read the input file and extract the basic information in the form of courses, rooms, teachers, curricula, constraints etc. The representation of our solution is in the form of Lectures (L). L's represents the total no of lectures of different courses. For each lecture(L) is schedule to a selected time slot and room (T,R) object where T is the timeslot assigned to lecture(L) within the given range from (1,..,..,TotalTimeSlot) and R represents the room range from (1,..,..NumberOfRooms). For each lecture (L) should keep track of Period(P), Room(R), Teacher(T), Course(C), Curricula, Available Period List, Room capacity and day. The proposed algorithm initiate by generating an empty initial solution in the form of two dimension Matrix (Days X PeriodsPerDay). Each period is divided on to total no of rooms.

4.1 Phase One of Proposed Solution

Phase one of proposed technique is used to generate course wise clusters with the help of Hierarchical Clustering Technique (HCT). HCT construct a set of nested clusters organized as a hierarchical tree. Two main types of hierarchical cluster are agglomerative and divisive. We are using an agglomerative clustering type in which it's start with the points as individual clusters and at each step merge the closest pair of clusters until only one cluster left. Traditional hierarchical cluster algorithms use a similarity or distance matrix. Different techniques are used to classify the distance between two clusters. we are using complete linkage clustering technique.

4.1.1 Generation of Similarity Matrix and Clusters

In similarity matrix total number of courses are define in horizontally and vertically. We have a small dataset Fis0506-2 (from ITC2007) with 30 courses. Now the algorithm counts that how many time courses comes with each other and puts its value in matrix. After putting all the courses values its merge the highest value courses with each other. The final similarity matrix is given below in table 4.1

Table 4.1: Final Similarity Matrix

	c0027 / c0028 / c0026 / c0083 / c0085 / c0079 / c0081 / c0082	c0038 / c0040 / c0045 / c0042 / c0044 / c0036 / c0034	c0006 / c0007 / c0009 / c0008	c0017 /c0018 / c0019	c0115 / c0116 / c0109	c0110 / c0111 / c0113	c0107 /c0108
c0027 / c0028 / c0026 / c0083 / c0085 / c0079 / c0081 / c0082							
c0038 / c0040 / c0045 / c0042 / c0044 / c0036 / c0034	0						
c0006 / c0007 / c0009 / c0008	0	0					
c0017 /c0018 / c0019	0	0	0				
c0115 / c0116 / c0109	0	0	0	0			
c0110 / c0111 / c0113	0	0	0	0	0		
c0107 /c0108	0	0	0	0	0	0	

The cluster of the similarity matrix are shown in fig. 4.1. where we have seven clusters, cluster one have eight, two have seven courses, three have four courses, cluster four, five and six contains three courses and cluster seven have only two courses.

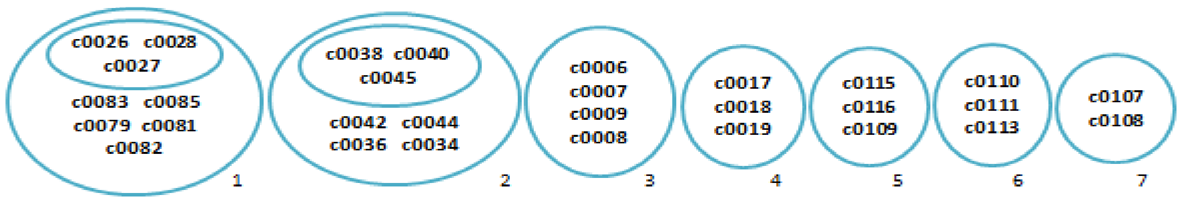

Figure 4.1: Clusters

4.1.2 Generation of Priority Table

After generating the cluster now we generate the priority table of all clusters. Priority table of dataset Fis0506-2 is given below in table 4.2. From Column one to four shows courses, lectures, Minimum working day and strength. Column five and six shows the depended courses and depended teacher courses of column one. No of Unavailability constraints shows in column seven, cluster no in column eight. We divide no of rooms of each course on 10, so the course which have schedule to less room have high priority its show in column nine. Column ten shows the rooms of each course in which it can be assign.

Priority value is generated by the following formula for each course.

Priority value = Total Lectures (TLec) + Total depended courses (TDC) + Average of Rooms (AR)
+ (Total working days - MWD) (TMWD) + Total unavailability constraint (TUC)

Table 4.2: Priority Table

Courses	Lectures	MWD*	Strength	Depended Courses		Depended Teacher Course	UAC*	Cluster	10 / Rooms	Rooms	Priority Value	Cluster Priority Average
c006	2	2	75	3	c0009 c0007 c0008	c0110	9	3	10.0 / 1 = 10	rC	27	28
c007	2	2	75	3	c0009 c0006 c0008	c0042	9	3	10.0 / 1 = 10	rC	27	28
c008	9	2	75	3	c0009 c0007 c0006	c0044	0	3	10.0 / 1 = 10	rC	25	28
c009	7	2	75	3	c0006 c0007 c0008	c0113 c0115	10	3	10.0 / 1 = 10	rC	33	28
c017	3	3	65	2	c0018 c0019	c0108	0	5	10.0 / 1 = 10	rC	17	21.3333
c018	9	2	65	2	c0019 c0017		0	5	10.0 / 1 = 10	rC	24	21.3333
c019	8	2	65	2	c0018 c0017		0	5	10.0 / 1 = 10	rC	23	21.3333
c079	6	4	11	3	c0026 c0027 c0028		0	1	10.0 / 5 = 2	rC rLUF2 rF rG rO	12	16.375
c081	6	4	8	3	c0026 c0027 c0028		0	1	10.0 / 5 = 2	rC rLUF2 rF rG rO	12	16.375
c082	5	4	11	3	c0026 c0027 c0028		0	1	10.0 / 5 = 2	rC rLUF2 rF rG rO	11	16.375
c083	5	4	11	3	c0026 c0027 c0028		0	1	10.0 / 5 = 2	rC rLUF2 rF rG rO	11	16.375
c085	5	4	11	3	c0026 c0027 c0028		10	1	10.0 / 5 = 2	rC rLUF2 rF rG rO	21	16.375
c026	6	4	52	7	c0085 c0027 c0028 c0081 c0079 c0083 c0082		9	1	10.0 / 2 = 5	rC rLUF2	28	16.375
c027	6	4	52	7	c0026 c0085 c0028 c0081 c0079 c0083 c0082		0	1	10.0 / 2 = 5	rC rLUF2	19	16.375
c028	3	3	52	7	c0026 c0085 c0027 c0081 c0079 c0083 c0082		0	1	10.0 / 2 = 5	rC rLUF2	17	16.375
c034	6	4	11	4	c0038 c0045 c0040 c0042		0	2	10.0 / 5 = 2	rC rLUF2 rF rG rO	13	15.5714
c036	6	4	11	3	c0038 c0045 c0040		0	2	10.0 / 5 = 2	rC rLUF2 rF rG rO	12	15.5714
c038	6	4	33	6	c0045 c0040 c0034 c0042 c0044 c0036		8	2	10.0 / 2 = 5	rC rLUF2	26	15.5714
c040	5	4	33	6	c0038 c0045 c0034 c0042 c0044 c0036		0	2	10.0 / 2 = 5	rC rLUF2	17	15.5714
c042	5	4	11	4	c0038 c0045 c0040 c0034		0	2	10.0 / 5 = 2	rC rLUF2 rF rG rO	12	15.5714
c044	5	4	11	3	c0038 c0045 c0040		0	2	10.0 / 5 = 2	rC rLUF2 rF rG rO	11	15.5714
c045	6	4	33	6	c0038 c0040 c0034 c0042 c0044 c0036		0	2	10.0 / 2 = 5	rC rLUF2	18	15.5714
c107	6	4	20	1	c0108		0	7	10.0 / 4 = 2.5	rC rLUF2 rF rG	10.5	13
c108	9	2	20	1	c0107		0	7	10.0 / 4 = 2.5	rC rLUF2 rF rG	15.5	13
c110	7	2	25	2	c0113 c0111		0	6	10.0 / 3 = 3.33	rC rLUF2 rF	15.3333	19.1667
c111	2	2	25	2	c0113 c0110		0	6	10.0 / 3 = 3.33	rC rLUF2 rF	10.3333	19.1667
c113	3	3	50	2	c0110 c0111		20	6	10.0 / 2 = 5	rC rLUF2	32	19.1667
c115	6	4	20	2	c0116 c0109		15	4	10.0 / 4 = 2.5	rC rLUF2 rF rG	26.5	17.1667
c116	4	3	20	2	c0115 c0109		4	4	10.0 / 4 = 2.5	rC rLUF2 rF rG	14.5	17.1667
c109	4	3	20	2	c0115 c0116		0	4	10.0 / 4 = 2.5	rC rLUF2 rF rG	10.5	17.1667

for course c006 we have a priority value 27 which comes

Priority value = TLec + TDC + TMWD + TUC + AR => 2 + 3 + (5-2) + 9 + 10 => 27

Cluster Priority Average decide which cluster is executed first so we need a cluster priority average. it is calculated by the following formula

CPA = Sum of all priority values in a cluster / Total number of courses in a cluster

4.2 Phase Two of TPC Technique

In the first phase of this technique all the courses are assign to different cluster and the priority is assign to each cluster. In each cluster every course have a priority value. Now all the courses are organized so in this phase the generation of timetable is done. The generation of timetable is starts from picking up the high priority cluster from priority table. After that highest priority course is selected from that cluster, the selected course is scheduled in timetable. In that time only HCV are satisfied such as rooms violation, teacher violation, conflicts violation etc. The algorithm of phase two is describe below

```
1   Extract basic information from files or database
2   Create Empty Solution Space
3   While select cluster with high priority
4      While select highest priority course from cluster
5         For select a Day
6             If Course Assignment Counter is equal
                  to Total Lectures of Course Then
7                Exit Day Loop
8            End if
9             For select a Period
10              if  Satisfied all Unavailability Constraints AND
                    Satisfied Room Constraints AND Teacher Availability
                    AND Curriculum Constraints then
11                  Course is assign to that time slot
12                  Increment Course Assignment Counter
13                  Exit For Loop Day
14              End if
15            Next period
16        Next day
17     Until all courses are schedule
18            Specific Improvement Moves
19  Until all clusters are finished
```

After the execution of above algorithm for first cluster the HCV free feasible timetable is generated for first cluster. There are number of SCV at this stage. To improve the SCV different improvement methods are used, so the penalty value of SCV is decrease and also HCV to remain zero. The terms moves means that a timeslots, courses, rooms etc are move or swap from one place to another for the purpose to decrease the SCV. The different specific improvement moves are used for CB-CTT are room move, lecture move, MWD move, room stability and capacity move and curriculum compactness move. After the top priority cluster is schedule then algorithm is move toward the second top priority cluster. This process is repeat until the courses of last cluster is adjusted to the timetable. A feasible and optimum timetable is generated at the end of procedure.

5.0 Experiential Results and Discussions

This section discuss the results of our Two Phase Cascading (TPC) technique for curriculum based course timetabling problem CB-CTP and compared it with Muller's hybrid technique (Unitime.org) because its approach won the International Timetabling Competition (ITC) that determined the limitations and constraint of CB-CTP (Di Gaspero, McCollum et al. 2007) also source code of muller's approach is provided.

5.1 Experimental Setup

Our proposed TPC (using c#) and Muller's techniques (using java) are executed on Core i3 computer with 2.67 GHz processor and 4 GB of RAM. For experiment we took twenty one benchmark data sets obtained by ITC competition (a real data from Italy University) (De Cesco, Di Gaspero et al. 2008). We executed both the techniques five times for every data set and have taken the best result out of five attempts. The results are evaluated using following criteria execution time , Hard Constraint Violation (HCV) and Soft Constraint Violation (SCV). The timetable are evaluated and tested by the standard c++ validator provided by the international timetabling competition (De Cesco, Di Gaspero et al. 2008).

The results produced by TPC and Muller's technique are shown in Table 5.1. It can be seen that the TPC execution time for all datasets is better than the muller's approach. For execution time our technique shows an average improvement of **13.93** for all twenty one datasets. Out of twenty one, eleven datasets shows a better result, three datasets generates the same results for both techniques and seven datasets shows the good results for Muller's. Our TPC technique generates a good timetable in term of both soft and hard constraint violation. Our technique shows an average improvement of **1.90** in all data sets in term of SCV over muller's technique.

Table 5.1: results produced by TPC and Muller's technique

Instance Name	**Muller's**			**TPC**			**Average Improvement in Time(Sec)**	**Average Improvement for SCV**
	HCV	**SCV**	**Time (Sec)**	**HCV**	**SCV**	**Time (Sec)**		
cmp01	0	5	12.3	0	**5**	**10.2**	**17.07**	**0**
cmp02	0	35	325.6	0	**24**	**298.3**	**8.40**	**31.42**
cmp03	0	66	168.4	0	72	**122.5**	**27.26**	-9.09
cmp04	0	35	88.4	0	38	**69.3**	**21.62**	-8.57
cmp05	0	298	1270.5	0	**287**	**1020.1**	**19.71**	**3.69**
cmp06	0	37	880.9	0	**28**	**645.2**	**26.76**	**24.32**
cmp07	0	7	1002.0	0	9	**857.1**	**14.47**	-28.57
cmp08	0	38	249.0	0	38	**222.3**	**10.73**	**0**
cmp09	0	100	256.3	0	**99**	**209.2**	**18.40**	**1**
cmp10	0	7	757.9	0	**6**	**745.6**	**1.62**	**14.28**
cmp11	0	0	15.2	0	**0**	**12.2**	**20.05**	**0**
cmp12	0	320	2502.1	0	**305**	**2150.1**	**14.07**	**4.68**
cmp13	0	61	283.7	0	69	**268.2**	**5.48**	-13.33
cmp14	0	53	242.3	0	**52**	**235.6**	**2.76**	**1.88**
cmp15	0	70	191.7	0	80	**162.2**	**15.39**	-14.28
cmp16	0	30	331.0	0	**28**	**298.5**	**9.81**	**6.66**
cmp17	0	70	448.3	0	79	**426.3**	**4.91**	-12.85
cmp18	0	75	295.2	0	**65**	**261.2**	**11.51**	**13.33**
cmp19	0	57	276.2	0	64	**229.3**	**16.98**	-12.28
cmp20	0	22	1201.2	0	**16**	**1014.2**	**15.57**	**27.27**
cmp21	0	89	356.3	0	**80**	**320.2**	**10.13**	**10.11**

Abdullah, S., et al. (2007). A hybrid evolutionary approach to the university course timetabling problem. Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, IEEE.

Abdullah, S., et al. (2012). "A hybrid metaheuristic approach to the university course timetabling problem." Journal of Heuristics **18**(1): 1-23.

Bellio, R., et al. (2012). "Design and statistical analysis of a hybrid local search algorithm for course timetabling." Journal of Scheduling **15**(1): 49-61.

De Cesco, F., et al. (2008). Benchmarking curriculum-based course timetabling: Formulations, data formats, instances, validation, and results. Proceedings of the Seventh PATAT Conference.

Di Gaspero, L., et al. (2007). The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3). Proceedings of the 1st International Workshop on Scheduling a Scheduling Competition (SSC 2007), Providence (RI), USA.

Jazzar, H. S. E. (2012). "Three-phase approach for curriculum-based course timetabling problem.(c2012)."

Kohshori, M. S. and M. S. Abadeh (2012). "Hybrid Genetic Algorithms for University Course Timetabling." International Journal of Computer Science Issues (IJCSI) **9**(2).

Lü, Z. and J.-K. Hao (2010). "Adaptive tabu search for course timetabling." European Journal of Operational Research **200**(1): 235-244.

MirHassani, S. and F. Habibi (2013). "Solution approaches to the course timetabling problem." Artificial intelligence review **39**(2): 133-149.

Müller, T. (2009). "ITC2007 solver description: A hybrid approach." Annals of Operations Research **172**(1): 429-446.

Qu, R. and E. K. Burke (2009). "Hybridizations within a graph-based hyper-heuristic framework for university timetabling problems." Journal of the Operational Research Society **60**(9): 1273-1285.

Wahid, J. and N. M. Hussin (2014). "Harmony Search Algorithm for Curriculum-Based Course Timetabling Problem." arXiv preprint arXiv:1401.5156.