International Journal of Basic and Applied Sciences, 4 (1) (2015) 44-50
www.sciencepubco.com/index.php/IJBAS

doi: 10.14419/ijbas.v4i1.4001
Research Paper

Formulation of low level heuristics

Aftab Ahmed [1]*; Abdul Raziq [2], Jan Muhammad [1], Manzoor Ali Brohi [2], Muhammad Abbas Khan [1]

[1] *Faculty of Information & Communication Technology, BUITEMS, Quetta*
[2] *Faculy of Management Sciences, BUITEMS, Quetta*
**Corresponding author E-mail: aftabshaikhs@gmail.com*

Abstract

The curricula scheduling is very significant and largely studied problem in academia. The desired solution calculatedly assembles the academic events over the carefully designed layout considering several predefined interlinked constraints. The contemporary research for solving scheduling constraints is inclined to raise the degree of generality, so that a wide range of identical problems may be addressed. The hyper-heuristic is such a state-of-the-art solving technique which stands on multi-layered framework. The top layer usually consists of classic algorithm for managing the operators on down-layers, and the same is occasionally assisted by machine learning or similar techniques. This research article examines the performance of the small group of bespoke low level heuristics. These LLHs are operated by hyper-heuristic to address the specific scheduling constraints. The set of heuristics are divided into a range of subgroups including timescale category which contain two subsets Day and Period. The utility group which contains two patterns named Shift and Swap techniques, while the third category encircles three more subgroups of Random or Sami-Random and Progressive.

Keywords: *Curricula Scheduling; Constraints; Low Level Heuristics.*

1. Introduction

The solution for curricula scheduling requires the assignments of the suitable venue (class-room or Lab) and personnel (Teacher, Lab Assistant etc.) over proactively identified transected point (slot) which also includes the multiple number of time variables (day, session and period) for different offered courses to students group (Curricula). The classic or manmade model essentially requires exhausting efforts only to obtain workable solution, on the other hand, along with other AI based computational models the Hyper-heuristic is the state-of-the-art solving technique and fairly capable to produce optimum results against exceedingly complex datasets. The set of low level heuristics (LLH) is unarguably a significant part of any hyper-heuristic framework. Usually the low level heuristics are designed according to domain specification. This article illustrates a small number of heuristics which is brief part of ongoing research project. In the said project the Low Level heuristics are categorized into several groups according to their relevant scope, functionality and process of interaction as shown in Table 2. The random low level heuristics apply to scatter the event variables over the properly formulated layout; as a result the some targeted slots can be vacated among others. Alternatively, incremental or progressive LLHs gradually try for positive changes and improve the efficiency or maintain its preceding state. The set of heuristics are supposed to make some progressive moves in datasets. This article is expansion of research work published by [1] and shown the functionally of four more important low level heuristics (LLH_8, LLH_9, LLH_{10}, LLH_{11}).

2. Related work

'Course (Curricula) timetable is integral part of academic calendar in order to commence the curriculum activities and later on regulate smoothly' [2]. The problem solution essentially depends upon the maximum satisfaction of the

constraints (Hard and Soft Constraints). 'Hard constraints [3] exceptionally should remain unbreakable under all circumstances. On the contrary all soft constraints cannot be satisfied in conventional problem instances although maximum satisfaction has decisive impact on solution feasibility'. Sometimes the two types of constraints are handled with distinctive solving approaches [2]. Designing data structures for dataset (description of events, resources and constraints) is significant step. 'An efficiently designed scheduling layout not just makes dataset comprehensible apparently, but greatly helps to converge error-free order of events. Each event-slot is a cross point between period and location. In that consequence, there is no chance of resources replication constraint e.g. Single class room allocation for two course'[4]. Computed initialization of dataset can be the part of solving hard constraints. Ahmed [5] implemented backtracking recursive algorithm for obtaining twofold tasks including placement of the Events and elimination of the Hard constraints. In the research article [6] introduces a novel evaluation function that accurately scans out soft and hard violations in the dataset and consequently assigns penalties to conflicting events.
Contemporary research direction in TTP is tending to raise the level of generality by state-of-art techniques so that a range of instances can be tackled. Hyper-heuristic is one of such modern-day techniques that largely shape such idea[7], [8].

3. Problem description

Curricula scheduling is a composite issue consists of several components such as Lecturer, Location, Course, Session, Duration and Students. So that formulation of Timetabling is TTS-Problem = [I, T, C, S, G]. The University Scheduling Problem hence is optimization of finite resources. Simply, the solution concludes the suitable location (Classroom) and resource personnel (Lecturer) on certain time-point (Day and Session) for the group of students (Curricula).

3.1. The problem terminology

- *Event:* An academic activity (Lecture) independently organized on distinct time.
- *Time-Slot:* Juncture point among resources (Lecturer or allocated Classroom) and time-point to hold the event.
- *Session:* stack of time-slots for running of several events.
- *Working Day*: A working day is combination of several sessions.
- *Physical Resource*: Classrooms, Buildings, Lab equipment, Personnel etc.
- *Group:* Enrolled students for the course(s)
- *Curricula:* Set of related optional or compulsory programs (courses).

3.2. Constraints penalty schemas

The penalty schemas are set of weights for constraints violation. The schema shares some global constraints and their penalties weight along with a few personalized constraints. These schemas belong to core performance measures. The sample penalty weights are reflected in Table 1, their raised level of complexity can be observed on each increasing scale. This penalty cost scheme is defined by Alex Bonutti et al [9]. In this work, one extra scale is added in all benchmark instances as shown in Table 1. Each Scale is started with a different list of penalty weights according to schemas. The evaluation function is multiple of the penalty cost with number of violation occurrences. The degree of complexity increases with the size of instance. The hard constraints (HCn) in Table 1 stand compulsory for feasibility level of solution. On the contrary, the satisfaction of soft constraints is significantly essential to obtains optimum results.

Table 1: Problem Classification Description[8].

Var.	Type	Constraint Label	Complexity Scales(1-6)					
H_1	Hard Constraints	Lectures	H	H	H	H	H	H
H_2		Conflicts	H	H	H	H	H	H
H_3		Room Occupancy	H	H	H	H	H	H
H_4		Availability	H	H	H	H	H	H
H_5		Room Suitability	-	-	4	H	-	-
S_1	Soft Constraints	Room Capacity	2	2	2	2	2	-
S_2		Min Working Days	6	6	-	2	6	6
S_3		Isolated Lectures	2	3	-	-	2	3
S_4		Windows	-	-	5	2	3	2
S_5		Room Stability	-	2	-	-	-	3
S_6		Student Min Max Load	-	-	3	2	3	2
S_7		Travel Distance	-	-	-	-	3	-
S_8		Double Lectures	-	-	-	2	-	-
S_9		Teaching Max Load	-	-	-	-	-	6

4. Methodology

The set of low level heuristics (LLHs) is essentially a core component of hyper-heuristic (HH) structure. Generally, LLHs are designed specifically for domain specification. Each LLH operates a small shift in current state of solution.

Table 2: Group of Classified Low Level Heuristics[1].

ID	Algorithm Name	Scope	Function	Interaction
LLH_1	Shift_MaxPenalizedDayToTail	Day	Shift	Semi –R
LLH_2	Shift_MaxSituratedDayToTail	Day	Shift	Semi –R
LLH_3	Shift_LessSituratedDay	Day	Shift	Semi –R
LLH_4	Shift_DayConstToLessSituDay	Day	Shift	Semi –R
LLH_5	Shift_WithDayConstImprov	Period	Shift	Progressive
LLH_6	Swap_InDays	Period	Swap	Semi –R
LLH_7	Shift_RandDayImprovment	Day	Shift	Progressive
LLH_8	Swap_InColumn	Period	Swap	Progressive
LLH_9	Shift_NeighboringPeroid	Period	Shift	Random
LLH_{10}	Swap_InRows	Period	Swap	Progressive
LLH_{11}	Shift_Dispersions_of _Exam	Day	Shift	Progressive

Table 2 exhibits the classification of Low Level Heuristics (LLHs) designed and developed in this research project, however last four LLHs are described in this article. The some LLHs are laying either in Shift or Swap functionality. The Shift based LLH moves the penalized data string to calculated vacant place whereas Swap function mutually interchange two events, this technique is useful in condensed problem instances. Random LLHs are used to throw the events faraway on the layout because the vacuities among the slots are required. Semi–Random category of LLHs picks the conflicting events out using predefined criteria and shift to diverse locations randomly. On the contrary, progressive LLHs in fact make positive move and improve or sustain the state. Figure 1 reflects the transaction of level of low level heuristics over problem instances.

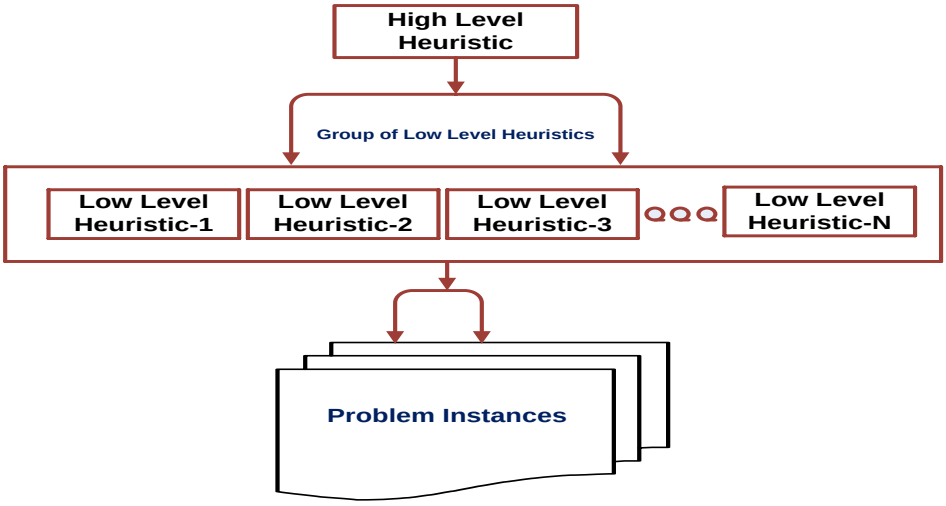

Fig. 1: Hyper-Heuristics Framework [1].

4.1. LLH_8: Swap in column

Figure 02 and LLH_8 algorithm illustrates a heuristic which is applicable in swapping of two events in scope of single session. The LLH is designed to address specifically some column level constraints for example the Room Constraint is very obvious target of this LLH. The LLH_8 is fundamentally a progressive operator, if it appears to be successful than outcome gets accepted otherwise the prior solution state becomes reinstated.

Algorithm LLH$_8$: Swap_InColumn

DEF Swap_InColumn(key1, 'Peroid'):

1. **FOR** k **IN** range (1, Rooms):
   a. **IF** Layout[k] **IS** None:
      key2 = k
   b. **ELIF** Layout[k]['Penlty'] != 0:
      key2 = k
   c. **ELIF** Layout[k]['PenType'] = = 'P':
      key2 = (key1 [0], key1[1],k)
   d. **ELSE:** key2 = None
2. **IF** key1 **AND** key2:
   a. SwapSlots(key1,key2)
   b. ResetFitness(Day)

Fig. 2: Exchanging Between in Single Column

4.2. LLH$_9$: Shifts in neighboring period

Fig 03 depicts the changeover of two penalized slots to adjacent period. It requires the empty slot in neighboring column. The LLH scans suitability of source and destination slots on penalty scale before executing the operator.

Algorithm LLH$_9$: Shift_NeighboringPeroid

DEF Shift_NeighboringPeroid(self):

1. key1 = self.ConstrintsType('**PERIOD**')
2. **FOR** n **IN** range (1,self.Rooms)
   Result = HC_threaten(NewPos)
   a. **IF** Result == 1: continue
   b. **ELIF** Result == 2: break
   c. **ELIF** Result == 0: key2 = NewKey
3. **IF** key2:
   a. ShiftFrom(key, key2)

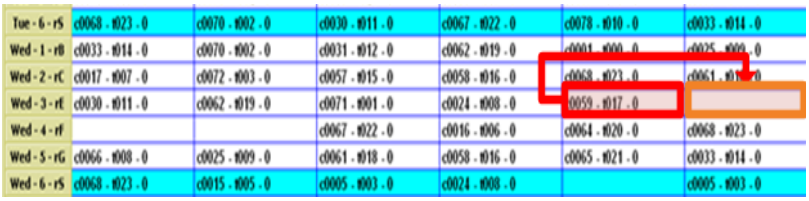

Fig. 3: Move to Adjacent Period

4.3. LLH$_{10}$: Swap in rows

The LLH$_{10}$ Algorithm and Figure 4 illustrate together the dynamic job of LLH$_{10}$ that calculatedly focuses breadth of layout for the constraints found in rows. The heuristic belongs to progressive group that ultimately ends with some improvements on the layout.

Algorithm LLH_{10}: Swap_InRows

DEF Swap_InRows():

1. key1 = self.ConstrintsType('Period')
2. **IF** key1 **IS NOT** None:
    a. HCFreeKeyList= EmpHCFreeDayKeys(key1,Swap =True)
3. **IF** len(HCFreeKeyList):
    a. **FOR** key2 **IN** HCFreeKeyList:
    b. SwapSlots(key1,key2)
    c. NowDaySum = self.SumPenDay(Day)
4. **IF** NowDaySum >= DaySum:
    a. Rollback (key2, key1)

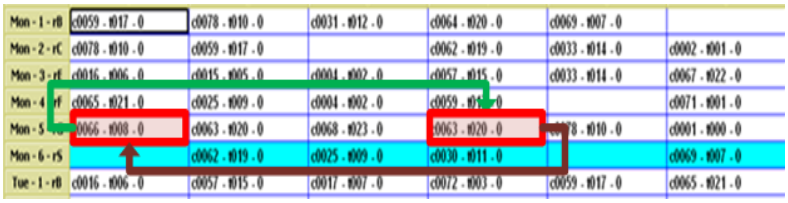

Fig. 4: Exchange of Slots In Row

4.4. LLH_{11}: Shift dispersions of exam

Heuristics LLH_{11} is gray shaded heuristics which might be used for exam and curricula timetabling. The same heuristic is discussed in [10] with PSO-Hyper heuristic for solving examination scheduling however in this study the said heuristic is member of curriculum solving LLHs, here the set is supervised by Genetic-Hyper heuristic. The LLH_{11} finds the difference between two timeslots, if there is no marginal gap than procedure moves the slot to less saturated day of the layout. The shifting process is inclined to increase in fitness function. Thus it operates on positive change if found in solution otherwise it calls the rollback action.

Algorithm LLH_{11}: Shift_Dispersions_of _Exam

DEF Shift_ Dispersions_of _Exam

1. (t_1,t_2) = ScanWeek()
2. **IF** $|t_i - t_{i+1}| \leq 1$
   Then
    a. Siturated_Day_key = MaxOfDay(t_1,t_2)
3. **IF** Less _Siturated_Day_key **NOT none**
   Then
    a. Less_Shift_DayToEnd(Siturated_Day_key)

5. Results

The partial experimental results of research prototype are showing the worth of research direction, this section briefly reflects them. The outcome is examined by a properly calculated performance measures. The project components are written and complied over Lenovo® Intel CORE ™ i3, 2.27 GHz, 4.00 GB RAM. The Python language version 2.7 and its supplement libraries are chosen to shape the project.

Table 3 exemplifies the two sample results of benchmark instances, which are grouped over six distinct complexity scales. Datasets are varying from each other on the basis of extended size of parameters including problem density (involvement of constraints categories), saturation (shortage of resources) and complexity level (increasing number of constraints). These results are generated by Hyper-Heuristic along with various other components as discussed in Related Work section and said heuristics $LLH_{8\text{-}11}$ were part of that compilation.

At the first the results reflects mutual and mandatory feature that the all the Hard Constraints are completely removed from layout and soft constraints are minimized to certain degree, however few of the soft violations remained unsolved in most of the cases.

Table 3: Comp03 (1-6) and Comp04 (1-6) Scales

Comp03	**Lectures**	**Conflicts**	**R Occupancy**	**Availability**	**R Suitability**	**R Capacity**	**MW Days**	**Isolated**	**Windows**	**R Stability**	**St Min Max Load**	**Travel Distance**	**Double Lectures**	**Teaching Max Load**	
	HC1	HC2	HC3	HC4	HC5	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8	SC9	Total
Sacle1	26	101	0	56		513	42	76							814
Unsolved	0	0	0	0		3	4	30							37
Solution	26	101	0	56		510	38	46							777
Per. Sol.	100.00	100.00	0.00	100.00		99.42	90.48	60.53							
Sacle2	26	101	0	56		513	42	76		423					1237
Unsolved	0	0	0	0		2	11	41		1					55
Solution	26	101	0	56		511	31	35		422					1182
Per. Sol.	100.00	100.00	0.00	100.00		99.61	73.81	46.05		99.76					
Sacle3	26	101	0	56	34	513			25		50				805
Unsolved	0	0	0	0	0	2			2		21				25
Solution	26	101	0	56	34	511			23		29				780
Per. Sol.	100.00	100.00	0.00	100.00	100.00	99.61			92.00		58.00				
Sacle4	26	101	0	56	34	513	42		25		50		5		852
Unsolved	0	0	0	0	0	313	0		2		12		0		327
Solution	26	101	0	56	34	200	42		23		38		5		525
Per. Sol.	100.00	100.00	0.00	100.00	100.00	38.99	100.00		92.00		76.00		100.00		
Sacle5	26	101	0	56		513	42	76	25		50	47			936
Unsolved	0	0	0	0		11	33	13	24		45	41			167
Solution	26	101	0	56		502	9	63	1		5	6			769
Per. Sol.	100.00	100.00	0.00	100.00		97.86	21.43	82.89	4.00		10.00	12.77			
Sacle6	26	101	0	56			42	76	25	423	50			10	809
Unsolved	0	0	0	0			4	7	6	10	3			3	33
Solution	26	101	0	56			38	69	19	413	47			7	776
Per. Sol.	100.00	100.00	0.00	100.00			90.48	90.79	76.00	97.64	94.00			70.00	

Comp04	**Lectures**	**Conflicts**	**R Occupancy**	**Availability**	**R Suitability**	**R Capacity**	**MW Days**	**Isolated**	**Windows**	**R Stability**	**St Min Max Load**	**Travel Distance**	**Double Lectures**	**Teaching Max Load**	
	HC1	HC2	HC3	HC4	HC5	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8	SC9	Total
Sacle1	20	103	0	51		84	37	78							373
Unsolved	0	0	0	0		1	1	19							21
Solution	20	103	0	51		83	36	59							352
Per. Sol.	100.00	100.00	0.00	100.00		98.81	97.30	75.64							
Sacle2	20	103	0	51		84	37	78		313					686
Unsolved	0	0	0	0		0	6	31		2					39
Solution	20	103	0	51		84	31	47		311					647
Per. Sol.	100.00	100.00	0.00	100.00		100.00	83.78	60.26		99.36					
Sacle3	20	103	0	51	34	84			29		57				378
Unsolved	0	0	0	0	0	1			1		2				4
Solution	20	103	0	51	34	83			28		55				374
Per. Sol.	100.00	100.00	0.00	100.00	100.00	98.81			96.55		96.49				
Sacle4	20	103	0	51	34	84	37		29		57		6		421
Unsolved	0	0	0	0	0	0	8		1		5		1		15
Solution	20	103	0	51	34	84	29		28		52		5		406
Per. Sol.	100.00	100.00	0.00	100.00	100.00	100.00	78.38		96.55		91.23		83.33		
Sacle5	20	103	0	51	34	84	121	78	29		57	19			596
Unsolved	0	0	0	0		0	44	0	12		24	9			89
Solution	20	103	0	51		84	77	78	17		33	10			473
Per. Sol.	100.00	100.00	0.00	100.00		100.00	63.64	100.00	58.62		57.89	52.63			
Sacle6	20	103	0	51			121	78	29	313	57			11	783
Unsolved	0	0	0	0			3	2	2	4	1			2	14
Solution	20	103	0	51			118	76	27	309	56			9	769
Per. Sol.	100.00	100.00	0.00	100.00			97.52	97.44	93.10	98.72	98.25			81.82	

6. Conclusions

In this research work, a subset of novel group of low level heuristics is elaborated. LLH(s) are addressing the multiple sets of benchmark datasets. All the LLHs are divided under three main categories. The outcomes reviled that each category have some targeted constraints where it can produce quality 'move' and it also demonstrate that solo category is not capable to produce sufficient level of outcomes however Hyper-heuristic managed various groups of LLHs promisingly produce optimal outcome.

Acknowledgment

This is the extended version of article [11] published as conference proceedings of ICCET 2013 held on March 5-7, 2013. In addition, as this study reflects only the small part of ongoing research project so some of the tables, figures and results are inherited from previous publications of same project in order to convey the notion properly.

References

[1] Aftab Ahmed, Mazhar Ali, Walayat Hussain, and A. H. S. Bukhari, "Bespoke Set of Heuristics for Solving Curriculum Scheduling Problems " *Sindh University Research Journal (Science Series)*, vol. 44, pp. 13-20, June 2012.

[2] Aftab Ahmed and Z. Li, "A Biphasic Approach for University Timetabling Problem," in *IEEE 2nd International Conference on Computer Engineering and Technology (ICCET 2010)*, Chengdu, Sichuan, China, 2010, pp. 192-197.

[3] Aftab Ahmed and Z. Li, "Solving Course Timetabling Problem Using Interrelated Approach," in *IEEE International Conference on Granular Computing San Jose*, California, USA, 2010, pp. 651-655.

[4] Aftab Ahmed, Mazhar Ali, Mirza Aamir Mehmood, and A. H. S. Bukhari, "Designing A Generic Layout For Acdemic Scheduling Problems," *Australian Journal of Basic and Applied Sciences*, vol. 5, pp. 1393-1397, 2011.

[5] Aftab Ahmed, Ghulam Ali Mashori, and W. Hussain, "Prograssive Dataset Initlization of Curiculum Benchmark Scheduling " *journal of Science, Technology, and Development*, vol. 30, pp. 7-15, 2011.

[6] Aftab Ahmed, Walayat Hussain, and A. Kamran, "Logic Formulation and Evaluation of Academic Constraints," *International Journal of Basic and Applied Sciences*, vol. 1, pp. 26-39, 2012.

[7] G. O. Jorge A. Soria-Alcaraz, Jerry Swan, Martin Carpio, Hector Puga, Edmund K. Burke, "Effective learning hyper-heuristics for the course timetabling problem," *European Journal of Operational Research*, vol. 238, pp. 77–86, 2014. http://dx.doi.org/10.1016/j.ejor.2014.03.046.

[8] Aftab Ahmed, Abdul Wahid Shaikh, Mazhar Ali, and A. H. S. Bukhari, "Hyper-GA for Solving Benchmark Scheduling Problems," *Australian Journal of Basic and Applied Sciences*, vol. 5, pp. 1657-1667, June 2011.

[9] Alex Bonutti, Fabio De Cesco, Luca Di Gaspero, and A. Schaerf, "Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results " *Annals of Operations Research*, vol. 179, 2010.

[10] Aftab Ahmed, Ahthasham Sajid, Mazhar Ali, and A. H. S. Bukhari, "Particle Swarm Optimizatin Based Hyper-Heuristic For Tackling Real World Examinations Scheduling Problem," *Australian Journal of Basic and Applied Sciences*, vol. 5, pp. 1406-1413, 2011.

[11] Aftab Ahmed, Riaz ul Amin, Muhammad Abbas Khan, and A. Sajid, "A Novel Set of Heuristics for Scheduling Constraints," in *3rd International Conference on Computer and Emerging Technologies (ICCET 2013) 2013.*