J. App. Em. Sc
Vol 5, Issue 1, July 2014

A Multilayered Heuristic for Solving Curricula Scheduling Problems

Aftab Ahmed[1], Muhammad Atif[2], Jamil Ahmad[2]

[1]Department of Computer Science, [2]Faculy of Life Sciences, Balochistan University of Information Technology, Engineering and Management Sciences, Quetta.

Abstract

Curricula Scheduling problem is recognized essentially on account of its vital significance in academia. The problem is echoed as tough resources placement job against troublesome constraints. The problem has been investigated by research community for several decades because of its inevitable importance and association with Non-deterministic Polynomial-time hard (NP-Hard) complexity. This research article investigates a novel and contemporary approach of using Memetic Algorithms (MA) centered Hyper Heuristic model to scrutinize the performance. The dynamic parameters of higher heuristic are get corrected and improvised with each iteration on the basis of performance measure. The signs learned from the experiments conclude the study-work steps forward in scheduling research and the scope of prospective and significant research direction are noticeable and remain open in the future. The work concluded with implementation of prototype coded in python language.

Keywords: *Heuristic, Curricula Scheduling, Constraints.*

Corresponding author's email: *aftab.ahmed@buitms.edu.pk*

INTRODUCTION

University scheduling problems are classified and planned into two very eminent formats, curricula and examination scheduling. These two types of academia scheduling possess variation in constraints and operate on different timescale but also share several of mutual constructive rules as well. Most probably, in all universities, opening of the semester and its conclusion depends on such planning & scheduling. In appearance, Curricula/Exam scheduling is a tabular form where academic activities generally are interconnected by available amount of human, physical or technical resources. To each work-day of weekly schedule comprises over *N* number of gatherings or periods recognized by fixed span of duration. Time sessions serve well-ordered piles of events whereas an event is set of interconnected information of course, set of students and lecturers, allocated precisely over timeslots and sites (Moscato , 1989).

There are two noticeable bunches of conditional rules called Compulsory (also known as Hard Constraints) and Discretionary (also known as Soft Constraints). The satisfaction of Compulsory Constraints is crucial requirement of solution whilst enough Discretionary Constraints steep up the overall performance. As the rule of thumb, the Compulsory Constraints are highly required not to be violated in any case, and considered mandatory qualification and very basic criteria for any solution. Discretionary Constraints are also extremely anticipated to be solved, however not essentially so. On most of the occasions, it is not possible to exterminate all soft violations. So as to frame the matter, dynamic penalty value is shaped to allocate constraint violations in order to measure the quality.

Due to the complexness of the problem extends (as a result of an outsized variety of events, enrolments, insufficient human and physical resources or massive bunch of constraints); solely an automatic timetabling system can produce optimal solution quickly as usually it is needed. Curricula planning & scheduling is an extremely tough topic in combinatorial optimization research, at the instant gets the interest of researchers from everywhere in the globe. A lot of research approaches are appeared for inspecting the

machine-driven course/exam timetabling for last many years. The research in work addresses the Memetic Algorithm (MA) investigation for design & developing examination scheduling problem. The study covers the plan and adaptive choosing mechanism of heuristic techniques with the purpose of solving challenging combinatorial search problems. Though numerous methods had been established up to now, the fundamental shortcoming of those schemes is the dearth of flexibility. Typically, Problem centered technique leads the whole process and concludes with compromising solution. The inspiration behind the study presented in this article is to raise the degree of generality upon which heuristics operators may work accommodatingly. (Burke, 1996)

Background

The Memetic are modified versions of standard Genetic Algorithms. The notion Memetic Algorithm was coined by Moscato (Moscato, 1989) in a Technical Report where he described an integrated heuristic which encompass Simulated Annealing and local search algorithm applied for agents game. Afterward, Moscato and Norman (Moscato, 1992) extended a similar approach that implemented local search within a Genetic Algorithm. In Burke et al.(1996) used a hill climbing local search as crossover operator that executed along with mutation operation. A couple of mutation operators were introduced follows the names LIGHT and HEAVY. A comparative study of proved that, approach obtained reasonably better results for the Nottingham capacitated examination Scheduling instance. Burke and Newall (1999) expanded the method in further research work that outperformed and produced promising results. The research results also demonstrated that encapsulated Genetic Algorithm local search approaches obtained better results than using the stand alone GA. Burke and Landa Silva (2004) exhibited in detail the blueprint of Memetic Algorithms for scheduling problems. Aftab et al. (2013) designed a novel parameter-based hyper heuristic approaches(s) for solving different benchmark (standard) scheduling problems. The dynamic parameters of low level heuristic selection are inclined to be tuned up on the basis of evaluation criteria. In each heuristic triggers a miner move on the basis of random and calculated manner (Aftab Ahmed et al., 2013; Aftab Ahmed et al., 2015). The LLHs are classified under several categories which reflects possible legal options available on Scheduler Events Container (SEC) (Aftab Ahmed et al., 2015).

Problem Description and Design

Curricula/Exam Scheduling can be planned by describing constraints (conditions), variables, their domains (range of values) and scores (violation penalties and reward for evaluation function). University scheduling schema may be disjointed into twofold type of constraints. The solving of Compulsory Constraints shape up the schedule on least possible applicability, whereas stretching fulfillment in Discretionary Constraints increases the reliability and optimality of solution. Generally, many of the Compulsory (Hard Constraints) are comprehensively practiced in all the universities; however some of the Discretionary (Soft Constraints) may differ from each one with reference to their educational priorities. In actual fact, a schedule is classically reflected merely workable if all the hard violations of the problem instance are removed.

Table 1 Set of Local Bespoke Heuristics

No.	ID	Heuristics Name	Scope	Function	Interaction
1	LBH_1	Move Max Penalized Day To Tail	Workday	Move	Random-H
2	LBH_2	Move Max Dense Workday To Tail	Workday	Move	Random-H
3	LBH_3	Move Less Dense Workday	Workday	Move	Random-H
4	LBH_4	Move Day- Constraint To Less Saturation Workday	Workday	Move	Random-H
5	LBH_5	Move With Workday Constraint Improvements	Session	Move	Incremental
6	LBH_6	Trade off In Workdays	Session	Tradeoff	Random-H
7	LBH_7	Move Random Workday Improvements	Workday	Move	Incremental
8	LBH_8	Trade off In Column	Session	Tradeoff	Incremental
9	LBH_9	Move Neighboring Period	Session	Move	Random
10	LBH_{10}	Trade off In Rows	Session	Tradeoff	Incremental
11	LBH_{11}	Shift Dispersion of Exam	Workday	Move	Workday
12	LBH_{12}	Trade off in Multiple Rows	Workday	Tradeoff	Incremental

Lower Layer – Local Bespoke Heuristics

Table 1 depicts the diverse features of Local Bespoke Heuristics (LBHs) considered in this research work. The LBH_{12} is an new heuristic added up in previous work (Aftab Ahmed et al., 2011) which extends the efficiency and effectiveness of ongoing project. Mostly, Local Bespoke heuristics executes either [Move] or [Tradeoff] process over penalized events. [Move] operation needs empty slot on destination side while [Tradeoff] operation is swapping of two academia events, particularly in highly saturated (dense) dataset instances. LBHs are planned to

lessen the overall penalty cost efficiently. The Random mode heuristics are accustomed to scuffle the events through Scheduler Events Container (SEC), thus the gaps may be implanted amongst the events for expedient shuffling. The Random-H kind of LBHs operators pull out the penalized events using some predefined rules and move to range of positions randomly. Alternatively, Incremental (progressive) LBHs essentially make few positive modifications into problem instance or recollect its previous state in case of failure. The scope of LBHs focuses the working range of subspace; some LBHs are operative on Day level and some on Session level.

Higher Layer of Hyper-MA Model

The research study inspects a Memetic Algorithm (MA) established over (Hyper-MA) for University Scheduling Problems (USTP). The Memetic Algorithm (MA) belongs to group evolution computation is imitation of famous Darwin's theory of evolution and an extended or modified version of Genetic Algorithm (GA). Memetic Algorithm (MA) gradually evolves the chromosomes (partial solutions) termed genome or specifically population. To each newly (descendants) produced population converges towards optimal outcome. Progenies for each successive generation are chosen under some conclusive set of rules. The cause of involving Memetic Algorithm (MA) method is because of its ripeness and capability to produce auspicious results. The Hyper-MA is applied to choose, execute and evolve a local heuristic from predefined set. The Hyper-MA is deployed on higher layer for appropriate selection and to concentrate generation where each chromosome is set of greatly fitted genes.

Algorithm 1: Hyper-MA Algorithm:

Sub-Procedure Hyper-MA Algorithm

Step (1):- Produce Tentative Solution S;

Step (2):- set $S^p : \leftarrow S$

Step (3):- **Repeat Loop**

i. Choosing Low Level Heuristics (LBHs)
ii. Apply **Adapted** Modifying Operators
iii. Apply Chrome of LLHs on Tentative Solution
iv. *fitness*(S^p) to each LBH
v. Produce Tentative solution S^{new}
vi. Simulated Annealing Technique
1. **CASE** S^{new} may be adopted **Than** set $S : \leftarrow S^{new}$
2. **ElseCase** $f(S^{new})$ **IsLessThan** $f(S^p)$ Then **SA-Criteria**(set $S : \leftarrow S^p$)

Step (4):- **Until** (Termination Schemas)

Step (5):- **return** Solution (S)

A Hyper-MA model is briefly demonstrated in Algorithm 1, which is extended and more efficient version in discussed (Aftab et al. 2013). The improved framework incudes the Simulated Annealing in modifying operators and solution rejection process. A Higher-level heuristic handles a group of Local Bespoke perturbative (incremental by alteration) heuristics. This implemented model is includes the whole research study. Initially, a preliminary solution (S) shaped than computing process move in in [Selection Phase] that choose a low heuristic (LBH_i) with higher fitness (to begin with random) designated and implemented on partially computed solution. Accordingly, new contender solution is produced as (S^{new}). Succeeding phase conforms the degree of quality solution by Evaluation Function (*fitness* (S^p)) to agree or disagree to the new move. If, it is accepted, the new candidate solution replaces the current one, else new move went through a SA-Criteria (set $S : \leftarrow S^p$) procedure so either it is accepted or thrown out and iterates again.

RESULTS AND DISCUSSION

The experimental results authenticate the accurate research course (direction) indeed. The computing methodology is inspected on benchmark (standard) datasets. Prominent outcomes discovered the comprehensive prospective and dimensions of adapted approach. Table (2) demonstrates the benchmark (standard) dataset which is categorized over six (6) divergent complexity scales. In this article, only one instance is reflected just for an idea of showing effectiveness and capability of proposed methodology, whereas several benchmark and real-world datasets are processed under the same project. The problem instances are diverged from one to other on the behalf of numerous eminent parameters containing density of the problem, saturation on site and complexity level etc. Scale1 comprises four (4) Compulsory Constraints (CC_1, CC_2, CC_3 and CC_4) and three (3) Discretionary Constraints in row (DC_1, DC_2 and DC_3) and number of constraints gradually increase with each complexity scale. Fitness functions carry an efficient piece of code for analyzing

the clashes against hard or Discretionary Constraints. CC_4 is perceived nullified (diminished) at the starting and keep same status during the entire course of computation.

Table 2: Benchmark Dataset

	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	
	CC1	CC2	CC3	CC4	CC5	DC1	DC2	DC3	DC4	DC5	DC6	DC7	DC8	DC9	Total
Sacle1	7	52	0	61		60	405	243							828
Unsettled	0	0	0	0		34	91	101							226
Settled	7	52	0	61		26	314	142							602
Per. Sol.	100.00	100.00	0.00	100.00		43.33	77.53	58.44							
Sacle2	7	52	0	61		60	405	243		264					1092
Unsettled	0	0	0	0		22	156	88		23					289
Settled	7	52	0	61		38	249	155		241					803
Per. Sol.	100.00	100.00	0.00	100.00		63.33	61.48	63.79		91.29					
Sacle3	7	52	0	61	19	260			250		330				979
Unsettled	0	0	0	0	0	111			70		112				293
Settled	7	52	0	61	19	149			180		218				686
Per. Sol.	100.00	100.00	0.00	100.00	100.00	57.31			72.00		66.06				
Sacle4	7	52	0	61	19	260	405		250		330		156		1540
Unsettled	0	0	0	0	0	81	52		0		53		71		257
Settled	7	52	0	61	19	179	353		250		277		85		1283
Per. Sol.	100.00	100.00	0.00	100.00	100.00	68.85	87.16		100.00		83.94		54.49		
Sacle5	7	52	0	61		260	405	243	250		330	150			1758
Unsettled	0	0	0	0		16	251	56	0		181	71			575
Settled	7	52	0	61		244	154	187	250		149	79			1183
Per. Sol.	100.00	100.00	0.00	100.00		93.85	38.02	76.95	100.00		45.15	52.67			
Sacle6	7	52	0	61			405	243	250	264	330			6	1618
Unsettled	0	0	0	0			5	46	3	44	5			2	105
Settled	7	52	0	61			400	197	247	220	325			4	1513
Per. Sol.	100.00	100.00	0.00	100.00			98.77	81.07	98.80	83.33	98.48			66.67	

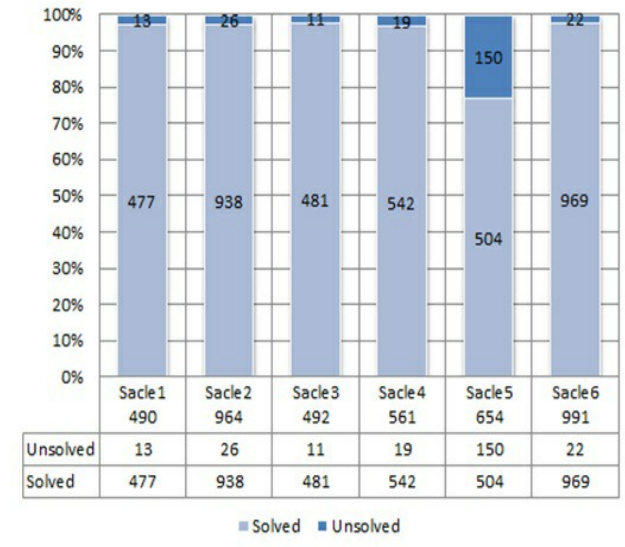

Figure 1: Benchmark Complexity Scale (1-6)

CONCLUSION

The study conducted in this research is planned to inspect the Memetic Algorithm based hyper-heuristic technique that evidenced to be outstandingly proficient of providing prominent results in an educational curricula scheduling problems. A hyper-heuristic is a recent and higher level of classy problem solving technique that implement a search over the space operated by a set of Local Bespoke heuristics (LBHs). The applied technique actually evolves the solver (operators) instead of solutions itself on an upper level of abstraction. Furthermore, the general model may simply be implemented on other instances of the similar class of the problem with trivial modifications. Research study is concentrating on effectual heuristic choice and move acceptance ways to choose the heuristics. The work presented here is an adaptation of the innovative methodology (Hyper-MA) containing twofold eminent segments. One of the future directions of the University Scheduling Problem is a multi-agent based model which would enable exceedingly parallel and distributed processing of problem and includes prototype of existing project.

REFERENCES

- Moscato P, On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. California Institute of Technology, California 1989.

- Moscato P and Norman MG. A Memetic Approach for the Traveling Salesman Problem Implementation of a Computational Ecology for Combinatorial Optimization on Message-Passing Systems. in *Parallel Computing and Transputer Applications*, Amsterdam., 1992, pp. 177–186.

- Burke EK, Newall JP and Weare RF. A Memetic Algorithm for University Exam Timetabling, in Burke and Ross. 1996, pp. 241–250.

- Burke EK and Newall JP. A Multistage Evolutionary Algorithm for the Timetable Problem, in IEEE Transactions on Evolutionary Computation. 1999, pp. 63–74.

- Burke E and Silva JL. The Design of Memetic Algorithms for Scheduling and

Timetabling Problems, in Recent Advances in Memetic Algorithms, Studies in Fuzziness and Soft Computing, 2004, pp. 289–312.

- Aftab Ahmed, Abdul Wahid Shaikh, Mazhar Ali, and A. H. S. Bukhari. (2011). Hyper-GA for Solving Benchmark Scheduling Problems. *Australian Journal of Basic and Applied Sciences.* 5:1657-1667.

- Aftab Ahmed, Riaz ul Amin, Muhammad Abbas Khan, and Ahthasham Sajid, A Novel Set Heuristics For Scheduling Constraints, in *3rd International Conference on Computer & Emerging Technologies (ICCET 2013)*, Shah Abdul Latif University, Khairpur, Sindh, 2013, pp. 179-183.

- Aftab Ahmed, Abdul Raziq, Jan Muhammad, Manzoor Ali Brohi, and M. A. Khan. (2015). Formulation of low level heuristics. *International Journal of Basic and Applied Sciences.* 4:44-50.