Comparison of Memetic Algorithm and Local Search Algorithm for University Exam Timetabling Problem(UTP)

Muskaan 2015CS10240

Divgian Singh Sidhu 2015CS10226

April 5, 2018

1 Introduction

1.1 The Timetable Problem

The scheduling of exams in institutions of higher education is known to be a highly constrained problem.Many constraints involved in exam scheduling vary from institution to institution.
Following constraints have been considered for this project:
1. *Hard constraint*: No student can sit more than one exam at any one time.
2. *Hard constraint*: For each period in timetable, the resource(rooms) demanded by the exams should not exceed the available resources.
3. *Soft constraint*: No student should have to take two exams in adjacent periods.

Conflicting exams are exams having common students.
Penalty defined as the number of students common between any consecutive periods.

1.2 Memetic Algorithm

Memetic algortihm uses hill climbing with genetic algorithm to improve its results.
Implemented this paper: https://link.springer.com/content/pdf/10.1007/3-540-61794-9_63.pdf for comparison with local search method.

1.3 Our Local Search Algorithm

Input:
1. Enrolements - a list of exams for each student.
2. Exam_time - list of exam and time duration of that exam.

3. Period_capacity - Period and its capacity according to the room sizes available.

Output:
1. Feasible Timetable
2. Conflicts
3. No. of Unscheduled exams

Algorithm 1:
1. Initialization:
- Choose exams randomly from unscheduled exams
- keep placing exams in period considering total capacity of period = sum of room capacities
//Not considering the constraint of room size

2. Hill climbing:
While no change in fitness of timetable is observed
    for every exam in timetable
    try placing it in another period with least penalty

Here, no. Of hill climbing iterations turns out to be approx = 10

3.Bin packing (next fit decreasing)
- Sort the rooms according to their capacity in decreasing order.
for every period
- Sort the exams in that period in decreasing order according to their capacity.
- Place exams starting from the first room wherever they fit.

In this implementation, room sizes were taken input and a coarse timetable was generated by not considering the room sizes in beginning and then with bin packing exams were fit in the rooms of periods.
Initialization was done differently in Genetic Algorithm by considering the room sizes.
So with this implementation, improvements after the hill climbing saturated and results were worse than Genetic Algorithm.
Due to unavailability of room sizes with datasets, period size(instead of individual room sizes) was consisdered in the final algorithm.

Algorithm 2:
1. Initialization:
- Choose exams randomly form unscheduled exams
- keep placing exams in period considering total capacity of period = sum of room capacities(or period capacity)
//Not considering the constraint of room of size

2. Hill climbing:
While no change in fitness of timetable is observed
    for every exam in timetable
    try placing it in another period with least penalty

Here, no. Of hill climbing iterations turns out to be approx = 10

3.**Local moves**
Following 4 moves were tried in a loop -
1. Pick an unscheduled exam and try scheduling it into any other period with minimum penalty.
2. If there exists no such period in which it can be scheduled without conflicts, then find a period which has minimum number of conlicting exams.
Try swapping all the conflicting exams with exams in other periods such that the unscheduled exam can be scheduled in that particular period.
3. If the exam is still not scheduled, then we schedule the exam in period with minimum(1) conflicting exams and unschedule the conflicting exam.
4. 2-swap in which every pair of scheduled exams are swapped if they reduce the penalty.

Local moves were tried independently but **Best Result** was obtained with Algorithm 2 by scheduling exams comprimising the penalty(soft constraint).

2 Results

Datasets taken from: http://www.cs.nott.ac.uk/~pszrq/data.htm
Dataset 1, 2 : www.cs.nott.ac.uk/~pszrq/files/instanceGenerator.zip
Dataset 3, 4 : ftp://ftp.mie.utoronto.ca/pub/carter/testprob

Observation 1 : From Table 1, it can be inferred that number of unscheduled exams were less than the result obtained from genetic algorithms.

Observation 2 : lp and sp datasets ahve different timetable density, which effects the result. In 17 / 20 results, local search performed better than genetic algorithms.

Observation 3 : In Burke's paper, it is claimed that "To give some comparison the Nottingham data was also used to test how well the algorithm compared against a straight-forward random descent algorithm. This consists of randomly generating a timetable then making random improvements until a number of tries have elapsed in which no improvements could be found.". But this local search algorithm gave promising results.

Observation 4 : It is also claimed that genetic algorithm takes respective time(1 hr, 1.5 hr) as mentioned in E.K.Burke's paper. Our algorithm for the same test case gave similar results in (0.5 hr, 0.5 hr). This can be seen from *Table 4*

3 Code

Code for implementation of genetic algorithms and local search is https://github.com/muskaankularia/local-search

				LS			LS+10 iter			MA	
Period	Exam	Room		Conflict	Unsched		Conflict	Unsched		Conflict	Unsched
10	88	500		76	48		89	46		122	48
15	140	500		223	57		236	54		215	58
20	189	500		245	68		247	64		351	68
25	215	500		397	66		416	57		382	65
30	283	500		591	52		542	48		419	63
35	327	500		621	47		680	44		644	52
40	358	500		656	44		736	35		606	52
45	416	500		823	26		815	21		668	37
50	463	500		991	17		922	14		868	21
55	487	500		884	11		887	11		965	11
30	289	200		357	1		349	1		360	7
30	267	250		431	15		453	8		250	13
30	264	300		456	23		509	19		460	22
30	278	350		486	28		501	25		550	37
30	280	400		527	41		526	34		530	39
30	283	500		591	52		517	50		520	55
30	282	600		461	70		491	65		480	74
30	274	700		488	81		583	75		560	84
30	284	800		629	85		548	86		530	100
30	274	900		431	96		493	91		450	100

Table 1: Varying number of periods keeping room capacity fixed and Varying Room capacity keeping number of periods fixed

			LS			MA	
Periods	Exams		UNSCHED	PENALTY		UNSCHED	PENALTY
15	80		0	0		0	0
15	100		0	0		0	0
15	80		0	0		0	0
15	80		0	1		0	0
15	80		0	36		0	42
15	80		0	86		0	76
19	100		1	113		2	119
19	81		1	56		0	74
19	80		1	96		0	100
20	526		0	26		0	110
20	511		15	498		40	425
24	508		17	681		44	505
30	533		26	820		57	616
35	542		39	1002		55	705
35	550		85	862		91	818
50	524		35	858		53	657
60	513		19	798		25	672
70	567		31	1028		46	714

Table 2: lp and sp dataset from www.cs.nott.ac.uk/~pszrq/files/instanceGenerator.zip

		LS				LS(PAPER)			MA		
Periods		Unsched	Penalty	time		Unsched	Penalty		Unsched	Penalty	time
32		0	0	36 min		0	0		0	0	1 hr
31		0	0	37 min		0	0		0	0	1.5 hr
30		0	25			0	4		0	4	
26		0	56			0	53		0	53	
23		0	252			3	269		0	269	

Table 3: Nottingham Dataset and comparison with paper's genetic algorithm results

	LS			MA	
	Unsched	Penalty		Unsched	Penalty
carf91	0	230		0	81
carf92	0	582		0	331
tre	0	31		0	3

Table 4: Other datasets from the same paper and comparison with paper results