CSE-318

(Artificial Intelligence Sessional)

Report on Offline-3 (Local Search)

(Exam Scheduling Problem)

Submitted by:

Tanjeem Azwad Zaman

Roll: 1805006

Dept. of CSE, BUET

Date of Submission:

23.01.2023

1.1 ***Introduction:***

Local search is a method for solving optimization problems that would otherwise be computationally hard. This is typically used on problems that can be formulated as finding a solution by maximizing /minimizing criterion among candidate solutions. Such algorithms move from solution to solution in the search spaceby making local changes until an estimated optimal solution is reached or, a time bound is elapsed.

1.2 Our Problem: Exam Scheduling

We are given courses and students enrolled in courses. There is a many to many relationship between them. The hard constraint states that we must schedule exams such that if 2 courses are scheduled on the same day then there is no student who is enrolled in both. Then the soft constraint asks us to minimize the pressure on students by introducing a notion of penalty.

1.3 Problem Formulation:

- Our problem can be formulated as a *graph-coloring, local search* problem as follows:
- Each node is a course with an edge between 2 nodes representing that a student is enrolled in both courses.
- We must assign colors to each node, where the color corresponds to an exam date. No 2 adjacent nodes can have the same color (hard constraint).
- After colors have been assigned, we try to minimize penalty

1.4 Constructive Heuristics: [Builds a feasible solution (non-conflicting exam scheduling)]

- **Largest degree**: The node with the largest number of edges (conflicting examinations) is scheduled first. Tie-break randomly.
- **Saturation degree**: Brelaz heuristic, prioritize nodes with more diversely colored neighbors, and then more uncolored neighbors for tie breaks.
- **Largest enrollment:** The largest number of students registered for the examinations is scheduled first.
- **Random ordering:** One randomly picked node (course) will be colored (scheduled).

1.5 Perturbative Heuristic: [reduces penalty (linear or exponential)]

- **Kempe-chain Interchange** : 1000 kempe chains of length greater than 2 were found and the color swapped.
- **Pair swap Operator :** colors of two randomly picked nodes were swapped and contraints checked for validity. 10000 such swaps were performed.

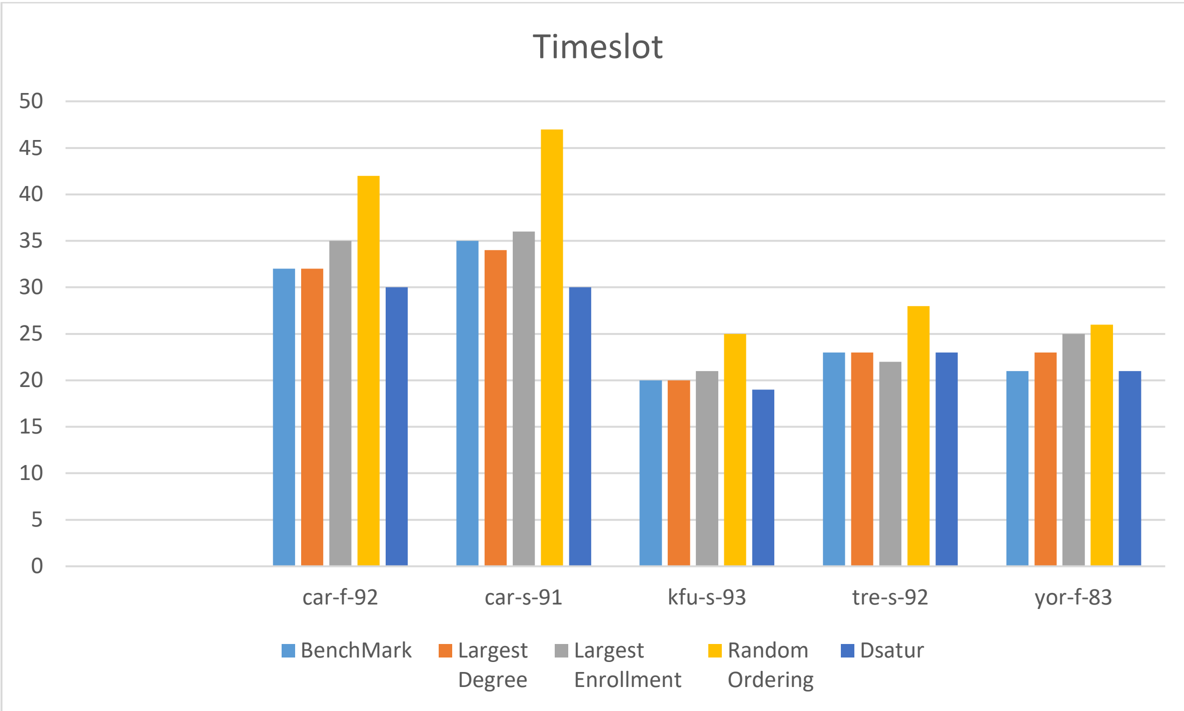
2. Tables:

Sl. No.	Benchmark Data	Known Best Solution		Scheme -1 Largest Degree + Kempe + Pairswap			
		Timeslots	Penalty	Timeslots	Penalty		
					Cons. Heuristic	Kempe	Pairswap
1	car-f-92	32	3.74	32	10.62	5.86	5.58
2	car-s-91	35	4.42	34	11.49	7.47	7.28
3	kfu-s-93	20	12.96	20	46.52	20.63	19.39
4	tre-s-92	23	7.75	23	15.89	10.69	10.38
5	yor-f-83	21	34.84	23	64.68	40.79	39.61

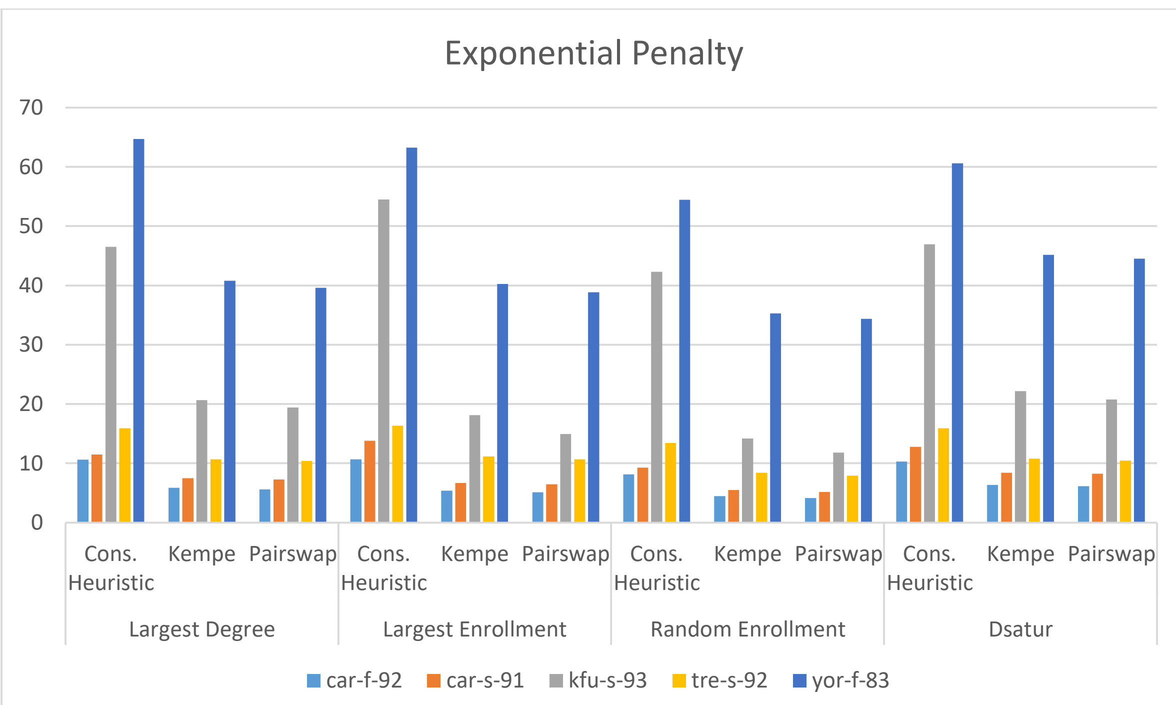
Sl. No.	Scheme -2 Largest enrollment + Kempe + Pairswap				Scheme -3 Random ordering + Kempe + Pairswap			
	Timeslots	Penalty			Timeslots	Penalty		
		Cons. Heuristic	Kempe	Pairswap		Cons. Heuristic	Kempe	Pairswap
1	35	10.69	5.39	5.1	42	8.12	4.43	4.14
2	36	13.78	6.66	6.44	47	9.24	5.47	5.16
3	21	54.51	18.11	14.93	25	42.31	14.16	11.82
4	22	16.32	11.17	10.69	28	13.4	8.42	7.92
5	25	63.27	40.24	38.84	26	54.43	35.27	34.37

Sl. No.	Scheme -4 DSatur + Kempe + Pairswap				Scheme -5 Linear + DSatur + Kempe + Pairswap			
	Timeslots	Penalty			Timeslots	Penalty		
		Cons. Heuristic	Kempe	Pairswap		Cons. Heuristic	Kempe	Pairswap
1	30	10.26	6.33	6.11	30	6.62	4.28	4.16
2	30	12.79	8.39	8.22	30	8.27	5.38	5.31
3	19	46.94	22.14	20.78	19	29.04	13.62	13.06
4	23	15.92	10.77	10.46	23	10.17	6.85	6.64
5	21	60.61	45.18	44.49	21	38.61	29.43	29.31

3. Charts:



Timeslot
50
45
40
35
30
25
20
15
10
5
0
car-f-92
car-s-91
kfu-s-93
tre-s-92
yor-f-83
BenchMark
Largest Degree
Largest Enrollment
Random Ordering
Dsatur



Exponential Penalty
70
60
50
40
30
20
10
0
Cons. Heuristic
Kempe
Pairswap
Cons. Heuristic
Kempe
Pairswap
Cons. Heuristic
Kempe
Pairswap
Cons. Heuristic
Kempe
Pairswap
Largest Degree
Largest Enrollment
Random Enrollment
Dsatur
car-f-92
car-s-91
kfu-s-93
tre-s-92
yor-f-83

4. Observations:

- The total number of timeslots is inversely proportional to the penalties incurred, for a given dataset.
- ***Scheme-4*** *(DSatur + Kempe Chain + Pairswap)* was the best in terms of *reducing the total number of timeslots*. This was followed by ***Scheme-1*** *(Largest Degree + Kempe Chain + Pairswap).* Consequently, these incurred the most penalties
- ***Scheme-3*** *(Random + Kempe Chain + Pairswap)* incurred the least penalties and thus had the worst timeslot counts.
- After the ***Kempe-Chain Reduction***, the ***Pairwise Swap Reduction*** returned very small optimizations in terms of penalties incurred.
- The Randomized heuristic would yield different results every time, and an average value was taken.

5. NB:

- 1000 iterations of successful reductions for Kempe-Chains of length >=3 were taken
- 10000 iterations of Pair-Swap attempts were taken.