Report on

Exam Scheduling using Local Search

Roll **1805093**

Introduction:

In this problem we were supposed to schedule examinations with least possible penalty using local search. The data in use was Toronto Dataset.

Our scope was specified within these benchmarks –

- Car-f-92
- Car-s-91
- Kfu-s-93
- Tre-s-92
- Yor-f-83

Constraints:

- **Hard**: No two exams of one student can overlap (happening in same day) each other. Every exam of a student must be scheduled on different day.
- **Soft**: Exams should be placed in such a way that a student can get the maximum day difference between any two exams

Penalty Strategy:

Linear Strategy: Let, n gap between any two exams

- If (n <= 5) penalty = 2 * (5-n)
- Else penalty = 0

Exponential Strategy: Let, n = gap between any two exams

- If (n <= 5) penalty = 2 ^ (5-n)
- Else penalty = 0

Heuristics:

- **Constructive:**
  - **Largest degree**: The node with the largest number of edges (conflicting examinations) is scheduled first. Tie break randomly.
  - **Saturation degree**: The well-known Brelaz heuristic (used in DSatur algorithm) provides dynamic variable (or node) ordering. (Refer to the wiki link for the algo)
  - **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). Nevertheless, you are free to devise any creative heuristic here instead of randomly picking up any node.

- **Perturbative:**
    - **Kempe chain Interchange:** A Kempe chain is defined as a connected subgraph that contains v, and that only comprises vertices colored with colors i and j. Take a particular Kempe chain and swap the colors of all vertices.
    - **Pair swap Operator:** A pair swap is the simultaneous application of two Kempe chain interchanges applied to Kempe (v, c(v), c(u)) and Kempe (u, c(u),c(v)).

      Let the Kempe chains KEMPE (u, i, j) and KEMPE (v, j, i) both contain just one vertex each (therefore implying that u and v are nonadjacent.) A pair swap involves swapping the colors of u and v. Used for building a feasible solution (non conflicting exam scheduling).

Results:

We ran every Constructive Heuristic followed by both of the Perturbative Heuristics with Exponential Penalty Strategy. After these 4 schemes, we ran Largest Degree Heuristic with both Perturbative Heuristics again but with Linear Penalty Strategy this time.

- Exponential Penalty

Benchmark	Known Best Solution		Largest-Degree + Kempe + Pairswap			
	Timeslots	Penalty	Timeslots	Penalty [After]		
				Largest-Degree	Kempe	Pairswap
car-f-92	32	3.74	32	10.38	6.94	6.87
car-s-91	35	4.42	35	11.6	9	8.95
kfu-93	20	12.96	20	46.5	35.84	26.54
tre-92	23	7.75	23	10.21	8.73	8.71
yor-83	21	34.84	23	65.53	49.15	48.56

Benchmark	Known Best Solution		Saturation Degree + Kempe + Pairswap			
	Timeslots	Penalty	Timeslots	Penalty [After]		
				Saturation Degree	Kempe	Pairswap
car-f-92	32	3.74	29	6.66	5.54	5.5
car-s-91	35	4.42	31	8.12	6.57	6.54
kfu-93	20	12.96	20	46.55	25.87	24.94
tre-92	23	7.75	22	16.01	13.83	13.81
yor-83	21	34.84	21	60.82	58.07	58.07

Benchmark	Known Best Solution		Largest Enrollment + Kempe + Pairswap			
	Timeslots	Penalty	Timeslots	Penalty [After]		
				Largest Enrollment	Kempe	Pairswap
car-f-92	32	3.74	35	10.67	6.78	6.76
car-s-91	35	4.42	36	13.78	8.33	8.32
kfu-93	20	12.96	21	54.52	25.76	23.75
tre-92	23	7.75	22	16.41	13.85	13.68
yor-83	21	34.84	23	40.69	31.55	31.24

Benchmark	Known Best Solution		Random Ordering + Kempe + Pairswap			
	Timeslots	Penalty	Timeslots	Penalty [After]		
				Random Ordering	Kempe	Pairswap
car-f-92	32	3.74	45	4.77	3.83	3.79
car-s-91	35	4.42	46	6.29	4.26	4.24
kfu-93	20	12.96	26	20.59	14.1	12.86
tre-92	23	7.75	28	8.46	7.28	7.06
yor-83	21	34.84	28	49.83	42.39	41.99

- Linear Penalty

Benchmark	Known Best Solution		Largest-Degree + Kempe + Pairswap			
	Timeslots	Penalty	Timeslots	Penalty [After]		
				Largest-Degree	Kempe	Pairswap
car-f-92	32	3.74	32	6.67	4.52	4.52
car-s-91	35	4.42	35	7.49	5.47	5.46
kfu-93	20	12.96	20	28.67	15.48	15.48
tre-92	23	7.75	24	10.12	8.97	8.33
yor-83	21	34.84	23	4.51	30.9	30.9

Discussion:

These heuristics were good to get closed to benchmarked values sometimes but the problem with them are none of them are consistent enough to give good results every time.