Report on

Exam Scheduling Using Local Search

Course ID: CSE 318

Course Title: Artificial Intelligence Sessional

Submitted By

Shehabul Islam Sawraz

1805088

Date of Submission: 24-01-2023

Introduction:

The exam scheduling problem is a combinatorial optimization problem that aims to schedule a set of exams and with a given set of constraints. The problem can be modeled as a graph coloring problem, where each exam is represented by a vertex, and two vertices are connected if the corresponding exams have common students. The goal is to assign a time slot to each exam, such that no two exams that have common students are assigned the same time slot, using the Local Search Algorithm. In short, we are supposed to schedule examinations with the 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 a 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) then penalty = 2 * (5-n)
- Else penalty = 0

Exponential Strategy:

Let, n = gap between any two exams

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

Heuristics:

- **Constructive:**
  - o **Largest degree**: The node with the largest number of edges (conflicting examinations) is scheduled first. Tie break randomly.
  - o **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)
  - o **Largest enrollment**: The largest number of students registered for the examinations is scheduled first.
  - o **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:**
  - o **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.
  - o **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 Data	Known Best Solution		Scheme - 1 Largest Degree + Kempe + Pairswap			
				Penalty		
	Time Slots	Penalty	Time Slots	After Largest Degree	After Kempe	After Pairswap
CARF92	32	3.74	32	10.597	7.126	6.875
CARS91	35	4.42	35	11.966	8.46	8.193
KFU93	20	12.96	20	46.63	25.84	21.77
TRE92	23	7.75	22	15.965	12.593	12.104
YOR83	21	34.84	23	63.643	47.087	45.841

Benchmark Data	Known Best Solution		Scheme - 2 Saturation Degree + Kempe + Pairswap			
				Penalty		
	Time Slots	Penalty	Time Slots	After Saturation Degree	After Kempe	After Pairswap
CARF92	32	3.74	30	9.577	7.665	7.638
CARS91	35	4.42	31	12.7	9.263	9.06
KFU93	20	12.96	19	46.757	26.518	23.272
TRE92	23	7.75	23	15.88	12.633	12.464
YOR83	21	34.84	20	61.65	54.488	54.294

Benchmark Data	Known Best Solution		Scheme - 3 Largest Enrollment + Kempe + Pairswap			
				Penalty		
	Time Slots	Penalty	Time Slots	After Largest Enrollment	After Kempe	After Pairswap
CARF92	32	3.74	34	10.69	6.84	6.47
CARS91	35	4.42	36	13.77	8.064	7.904
KFU93	20	12.96	21	54.56	24.5	20.17
TRE92	23	7.75	22	16.323	13.387	13.142
YOR83	21	34.84	24	63.98	44.55	42.78

Benchmark Data	Known Best Solution		Scheme - 4 Random Ordering + Kempe + Pairswap			
				Penalty		
	Time Slots	Penalty	Time Slots	After Random Ordering	After Kempe	After Pairswap
CARF92	32	3.74	34	10.945	6.92	6.62
CARS91	35	4.42	36	13.68	8.12	7.7
KFU93	20	12.96	21	54.535	25.233	19.768
TRE92	23	7.75	23	16.48	12.312	11.943
YOR83	21	34.84	24	63.982	44.302	43.533

- Linear Penalty

Benchmark Data	Known Best Solution		Scheme - 1 (Linear Penalty) Largest Degree + Kempe + Pairswap			
				Penalty		
	Time Slots	Penalty	Time Slots	After Saturation Degree	After Kempe	After Pairswap
CARF92	32	3.74	32	6.75	5.03	4.82
CARS91	35	4.42	35	7.62	5.98	5.79
KFU93	20	12.96	20	28.77	18.557	15.82
TRE92	23	7.75	23	10.19	8,75	8.54
YOR83	21	34.84	23	40.33	31.596	30.359

Discussion:

These heuristics are good to get close to benchmarked values sometimes but the problem with them is that none of them are consistent enough to give good results every time. It is also observed that as the total number of time slots decreases, the penalty calculated by the exponential approach increases. This indicates that there is a trade-off between the number of time slots and the number of conflicts between exams of the same student.