

Exam scheduling using local search

CSE 318



January 23, 2023

MD. ZARZEES UDDIN SHAH CHOWDHURY

1805009

**Task:**

Designing an exam timetable of students using local search,  
given all the ids of the courses taken by each student. In the end, getting a day-wise plan of exam scheduling of the courses.

**Goal:**

Primary goal is to minimize the number of timeslots needed for  
the problem. Secondary goal is to minimize the penalty amount of  
the scheduling.

**Equipment:**

* All the tests are done in a core i3, 4GB RAM laptop

Benchmark Datasets:

* car-f-92
* car-s-91
* kfu-s-93
* tre-s-92
* yor-f-83

**Procedure:**

Using 4 constructive heuristics number of timeslots were determined. Then penalties were reduced by two perturbative heuristics applied consecutively. In first 4 tables, penalties were calculated using exponential formula. In the last table linear formula was used. As instead of random heuristic, we were free to devise any creative heuristic, I applied normal BFS out of curiosity.

**Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Benchmark data | Known best solution | | Scheme-1  Largest Degree + Kempe + Pair Swap | | | |
| Time slots | Penalty | | |
| Timeslots | Penalty | After Largest Degree | After Kempe | After  Pair  Swap |
| CARF92 | 32 | 3.74 | 32 | 10.605 | 5.203 | 4.798 |
| CARS91 | 35 | 4.42 | 35 | 11.748 | 6.371 | 5.835 |
| KFU93 | 20 | 12.96 | 20 | 46.443 | 17.757 | 16.412 |
| TRE92 | 23 | 7.75 | 22 | 15.668 | 10.731 | 10.462 |
| YOR83 | 21 | 34.84 | 23 | 64.684 | 35.950 | 35.221 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Benchmark data | Known best solution | | Scheme-2  Saturation Degree + Kempe + Pair Swap | | | |
| Time slots | Penalty | | |
| Timeslots | Penalty | After Saturation Degree | After Kempe | After  Pair  Swap |
| CARF92 | 32 | 3.74 | 29 | 10.077 | 5.927 | 5.594 |
| CARS91 | 35 | 4.42 | 31 | 12.219 | 7.392 | 6.941 |
| KFU93 | 20 | 12.96 | 19 | 46.941 | 19.047 | 17.812 |
| TRE92 | 23 | 7.75 | 23 | 15.843 | 9.919 | 9.571 |
| YOR83 | 21 | 34.84 | 21 | 62.042 | 43.932 | 43.390 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Benchmark data | Known best solution | | Scheme-3  Largest Enrollment + Kempe + Pair Swap | | | |
| Time slots | Penalty | | |
| Timeslots | Penalty | After Largest Enrollment | After Kempe | After  Pair  Swap |
| CARF92 | 32 | 3.74 | 35 | 10.696 | 4.463 | 4.119 |
| CARS91 | 35 | 4.42 | 36 | 13.780 | 5.879 | 5.371 |
| KFU93 | 20 | 12.96 | 21 | 54.477 | 14.036 | 12.895 |
| TRE92 | 23 | 7.75 | 22 | 16.825 | 10.468 | 10.170 |
| YOR83 | 21 | 34.84 | 24 | 63.869 | 34.177 | 33.614 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Benchmark data | Known best solution | | Scheme-4  Normal BFS + Kempe + Pair Swap | | | |
| Time slots | Penalty | | |
| Timeslots | Penalty | After Normal BFS | After Kempe | After  Pair  Swap |
| CARF92 | 32 | 3.74 | 39 | 9.325 | 3.977 | 3.529 |
| CARS91 | 35 | 4.42 | 41 | 12.149 | 4.885 | 4.375 |
| KFU93 | 20 | 12.96 | 23 | 37.604 | 13.409 | 11.562 |
| TRE92 | 23 | 7.75 | 28 | 14.598 | 7.541 | 6.967 |
| YOR83 | 21 | 34.84 | 25 | 61.234 | 32.775 | 31.676 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Benchmark data | Known best solution | | Scheme-5  Largest Degree + Kempe + Pair Swap | | | |
| Time slots | Penalty (Linear) | | |
| Timeslots | Penalty | After Largest Degree | After Kempe | After  Pair  Swap |
| CARF92 | 32 | 3.74 | 32 | 6.762 | 3.369 | 3.143 |
| CARS91 | 35 | 4.42 | 35 | 7.518 | 4.122 | 3.849 |
| KFU93 | 20 | 12.96 | 20 | 28.696 | 11.560 | 10.812 |
| TRE92 | 23 | 7.75 | 22 | 10.034 | 7.013 | 6.812 |
| YOR83 | 21 | 34.84 | 23 | 40.837 | 23.169 | 22.982 |

**Discussion:**

Using constructive heuristic, the number of time slots were determined, though those heuristics doesn’t guarantee the most optimal solution rather finding a solution quickly. Perturbative heuristictries to minimize penalty, again doesn’t guarantee the most optimum. In spite of these issues, we are getting solution close to best due to the nature of local search. A local search algorithm starts from a candidate solution and then iteratively moves to a neighbor solution; a neighborhood being the set of all potential solutions that differ from the current solution by the minimal possible extent. This requires a neighborhood relation to be defined on the search space. Here Kempe chain and pair swap heuristics are example of hill climbing and iterative local search.

In Scheme-2(Saturation degree + Kempe + Pair Swap) number of time slots are comparatively lower than other schemes, but penalty amount is higher. In Scheme-4(Normal BFS + Kempe + Pair Swap), penalties are at the lowest but timeslots are highest. Scheme-1(Largest Degree + Kempe + Pair Swap) and Scheme-3(Largest Enrollment + Kempe + Pair Swap) give better results than the others with respect to both timeslots and penalty. Among those two, largest degree heuristic gives close to known best solution.

**Result:**

* Scheme-1 is the best heuristic here. (Largest degree + Kempe chain + Pair swap)
* Scheme-3 is the second best. (Largest enrollment + Kempe chain + Pair swap)