|  |  |  |  |
| --- | --- | --- | --- |
| Code | Genre | Description | Quotes & Key Points |
| Werra1985 | Background | A historic literature review at popular approaches pre-1985 | * Interactive models are important * Node colouring in a graph is popular * Can produce a network for exam graphs, which is smaller than course scheduling designs |
| Goulas2020 | Background | Investigating factors that affect exam performance. Very useful for defining soft constraints. | * Scheduling affects are more pronounced for STEM exams * Positive relationship between exam score and exam order for STEM * Negative relationship between exam score and number of days since first exam for STEM * Optimizing schedule can improve performance by 0.02 standard deviations * Cognitive fatigue. A student who takes an exam 24 days after first exam experiences a decrease in performance by 0.14 of a SD. * Taking an exam one place later in order (i.e. 5th exam over 4th) increases performance in STEM subjects by 0.016 of an SD * Higher number of exams taken gives higher benefit from optimizing exam scheduling |
| Pietro2012 | Background | Investigating how switching from semester to yearly exams affects student performance | * Statistically negative affect of exams being placed at the end of the academic year * Could be because students suffer high levels of stress * A long gap between teaching and exam may decrease performance levels |
| Wear1997 | Background | A review of popular scheduling techniques in 1997 | * A timetable is a placement of a set of meetings in time * Exams can share rooms, and can be split across rooms. * 42% of universities did not use a computer for scheduling * Genetic representation of timetable is a long string encoding when and where meetings take place * Memetic algorithms expand on Gas * Simulated annealing – keeps track of a feasible timetable. After each iteration, a new feasible timetable is produced. Very similar to Tabu search |
| Gashgari2018 | Background | Survey of approaches in 2018 \*\*\* contains a nice table of pros and cons of different approaches \*\*\* | * General approaches are: Direct Heuristics (from graph colouring),MILP, GA’s, Simulated Annealing, Tabu Search, Quadratic assignment |
| Pope2015 | Background | Investigation into how time between cognitive tasks effects performance | * Student taking two exams does significantly better when further apart * Relationship is linear in 0-11 days * Most of the effect is concentrated on the second exam |
| Burke2005 | Genetic Algorithms | Looking at genetic algorithms | * Hybrid approaches can yield even better results than traditional evolutionary techniques * Evolutionary algorithm with local search is known as a memetic algorithm * Two fundamental constraints: no entity can be demanded to be at more than one place at a time. And we cannot exceed resource availability * This simple problem could be solved using heuristic assignment. * Common second constraints: (no students have exam on two adjacent periods), (no two exams same day), (exam a must be before exam b), (exam A and B must be at same time), (exam must be in particular room) * Memetic algorithms reduce the space of possible solutions to the subspace of local optima * Hill climb algorithm: (take each period, then each event. Now try to place e in a different period, and pick the one which causes least penalty). This is done on unscheduled events. |
| Cheng2004 | Pure Methods |  | * The problem is to generate schedules that satisfy hard constraints while minimizing soft constraints * Soft constraints: avoid 3 consecutive exams, avoid having only one exam in a day, avoid students having an exam in last slot of day * First stage: assign events to rooms so that all events are assigned, and no student has two exams at same time. * Construct a graph: nodes are exams, and edges are one if two students share that exam. Weight function is defined, counting the number of instances of common instances between two exams for a given exam. Count edges out from the vertex * Chooser vertices with high degree first. Highly conflicting events assigned first helps * Look at weighted bipartite matching * Very fast solution |
| Dener2018 | Genetic Algorithms | Looking at a two stage genetic algorithm. | * Try to minimize length of exam schedule * Soft constraint: not meeting room capacity * Nice flow chart * GA is effective for scheduling |
| Mansour2007 | Genetic Algorithms | Looking at SA and GA. | * Scheduling done manually can be unfair * Colour the vertices of the graph using a specified maximum number of colours |
| Shatnawi2017 | Genetic Algorithms | Looking at GA | * Soft constraints: spread exams as far as possible, large number of student exams should be set first to help marking |
| Gedeon | Genetic Algorithms | Looking at GA | * Experimentally, subjects with fewer clashes were being scheduled sooner than exams with large number of clashes * This causes GA to decrease very rapidly, then slow down * Took 8 hours to do 81 exams for 2823 students |
| Corne1993 | Genetic Algorithms | Good explanation of GA | * Always found a better timetable than the one used, and always took less than 30 minutes * Gas are good at finding global optima in very hilly spaces * Randomly generate our chromosomes. Evaluate each chromosome in the generation, with the sum of scores denoted S. Then select P/2 pairs, with probability proportional to the fitness. For each pair use a recombination operator which gives two child chromosomes from the parents. Use a mutation on each child * This selection is known as roulette wheel selection. * Chromosome [2,4,7] represents exam 1 in slot 2, exam 2 in slot 4 * More than one exam at once (30), more than two exams in one day (10), two exams in consecutive timeslots (3), exam before and after lunch on same day (1) |
| Weitz1997 | Heuristics | Comparison between 3 heuristics. | * Forming maximally diverse groups (based on multiple criteria) of a specified size for a given population is a mathematically equivalent problem * If edge weights satisfy triangle inequality, value of optimal solution is at most 2(s-1)/s * Select random student. Next student is the “least different” from the first student, they are placed in next group. This continues. This is cheap, so you could run this for each of the N students, and pick best solution manually. (THE WJ Heuristic) * Begin by creating an arbitrary initial solution. For this assignment, determine R which is a matrix specifying for all students and groups, the contribution to the total difference that is the result of the student being assigned to a different group. Look at this heuristic (LCW) Lofti-Cerveny Weitz |
| Malik2007 | Heuristics |  | * 73% of students prefer exams scheduled in same room * Disturbance effects such as student movements happen when different exams are scheduled in the same room |
| Zhaohui | Heuristics |  | * Another hard constraint: all examinations should be in only one venue * Maximize number of sessions between exams for each student * This is a multiple knapsack problem * Perform a greedy colour of the graph. Then group the vertices by colour. Re-order the vertices in this group according to their group, so that all vertices in each group are consecutively ordered |
| Glaser2022 | Background | Looking at how exam time affects performance | * A day long break in the middle of exam period significantly increases average grade * Morning exams are 20-25% lower than afternoon * Fatigue results in lower marks as the exam period proceeds |
| Sansani2019 | Background |  | * No affect of having more days free before an exam * Benefits of a shorter schedule make costs less for university |
| Elliot1999 | Background | Discussion of how to study for exams |  |
| Davidson2001 | Background |  | * Best predictors of performance are prior marks and motivation for taking the course |
| Davis1980 | Background |  | * Students expect to perform well on exams, and when they do not they avoid blaming themselves |
| Meseguer2006 | Background | Soft constraints | * Typically use a search in the space of all solutions * Gives us a way of discriminating between all potential solutions that satisfy hard constraints * Weighted constraints (soft constraints that bare a penalty) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |