Timetabling with Genetic Algorithms

Timetabling Problem

- Specifically university class timetabling
- Highly complex problem (NP-Hard)
- Example: School of Computing
  - Hundreds of interactions between students and lecturers per week
  - Complex set of constraints
  - Fixed number of rooms and timeslots (~50 rooms on different sites, 40 hours in a week)

Timetabling Problem

- Approaches include Tabu Search, Tiling Algorithms, Simulated Annealing and Multi-Agent Systems
- GAs a good candidate
  - Work well on other scheduling tasks
  - Have previously been applied to timetabling in several different ways

Constraints

- Hard constraints (weight 1.0) – if broken result in invalid timetable
  - All classes must be scheduled
  - No class/lecturers double booked
  - Room capacities must not be exceeded and correct room types used
- Soft constraints (weight 0.01) – relate to quality of a feasible timetable
  - Classes scheduled within preferred hours
  - Hour for lunch is allowed
  - Bunch classes into groups
  - Avoid long runs of consecutive lectures

Fitness Function

- To use GAs or MAs for timetable generation, we need a way of evaluating a timetable
- Timetable fitness:

$$f = \frac{1}{1 + \text{ConstraintViolations}}$$

Timetabling GA 1

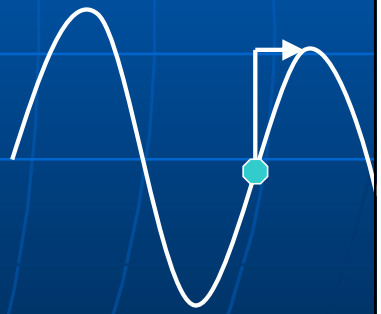
- Two approaches studied
- In one, each value (allele) in chromosome represents the timeslot and room given to a module
  - So allele value 0 means room 0, Monday, 9am
  - Value 39 means room 0, Friday, 4pm
  - Value 40 means room 1, Monday, 9am
  - …
- Large number of allele values, heavy onus on fitness function

Timetabling GA 2

- In the second GA, each allele represents the timeslot assigned to a class
- A greedy algorithm assigns the rooms later, taking the classes in order of size and assigning rooms to each in turn

Memetic Algorithms (MAs)

- Builds on the idea of a GA
- GAs have static genes being passed through generations, MAs have memes which can be changed
- Adds local search (hillclimbing) to algorithm; aims to reduce search space MA must cover



Memetic Algorithms (MAs) cont.

- Could adapt either of the GAs
- Initially tried both, had to drop one due to time constraints
- The timetabling MA adds a local search to the genetic + greedy algorithm hybrid

Local Search

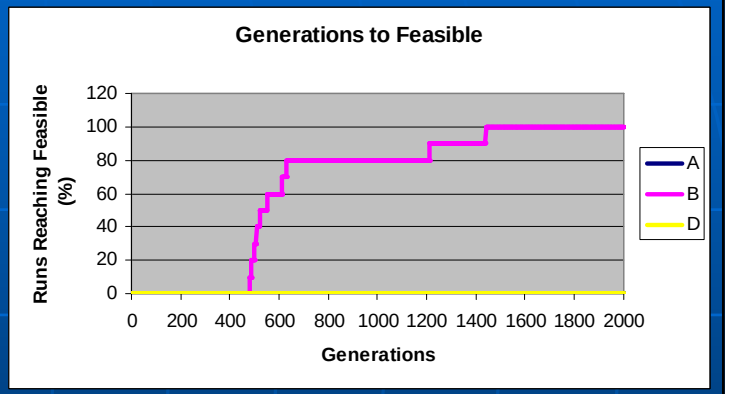
- Attempts to resolve problems with the timetable
- Takes classes which clashed with others and attempts to find a new timeslot where there is no clash

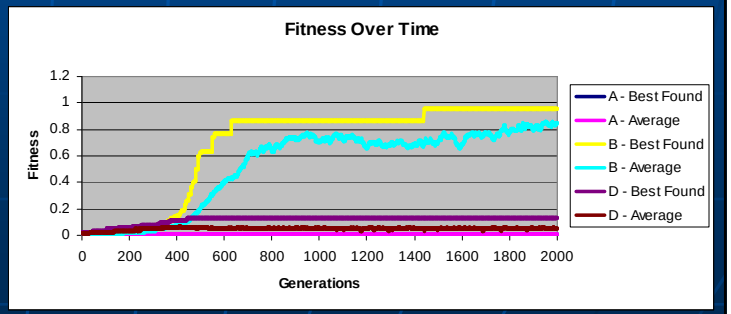
Optimisation

- Fractional factorial screening experiment determines significant factors
- Response surface experiment finds optimal values for those factors
- Confirmation experiment run to check values

Comparison

- Comparing number of generations to find a feasible timetable, and fitness over time
- Found hybrid GA to be best approach
- Faster and better quality solutions
- MA performed surprisingly poorly – possible local search implementation issue





Conclusions and Future Work

- Successfully confirmed that timetabling can be automated with GAs
- More work required on MA
- Include extra "features" of the School timetable
  - Electives
  - Classes shared with other schools
  - Account for distance between buildings

Cohort: Foundation Course Group A,28

	0	1	2	3	4	5	6	7
Day 1	CM1005, L, C47		CM1003, LAB, C8/C8A					
Day 2	CM1004, L, C47	CM1004, LAB, C8/C8A						
Day 3						CM1010, LAB, C8/C8A		
Day 4	CM1005, LAB, C8/C8A						CM1010, L, C47	
Day 5	CM1005, T, C47	CM1003, L, C47	, T, C47	CM1010, T, C47				CM1003, T, C47

Questions?