جامعة بيرزيت

BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Electrical and Computer Engineering Department

ARTIFICIAL INTELLIGENCE

PROJECT #1

Instructor Name: Aziz Qaroush

Student Name: Amjad Shomali
Student ID: 1161957

16/10/18

Formalization:

Initial state: a random, heavily conflicting, graduation project discussion schedule

Final state: A schedule with 0 conflicts being:

- Each instructor has one project each time slot
- Instructor has at least 3 and max 6 project discussions
- discussions are balanced between instructors
- no instructor has a 3 consecutive discussions

Gene: Object consisting of project id and title, students, supervisor and examiners

Chromosome: An array of object arrays, meaning if we have 5 time slots a day, 5 parallel each timeslot we'll get an array of length 10 (slots a day * days) each array index has an array of 5 genes

[[g1,g2,g3,'N/A',g4] , [g5,'N/A',g6,'N/A','N/A'] …. ->

N/A being an empty timeslot or a no-gene

Then the times are generated based upon the gene-group positions and given inputs

Procedure:

Input data is read as a pattern out of two excel sheets, one containing the projects/students/supervisor/preference combination and another containing available examiners and their preference

Genes are generated by pairing random but distinct examiners (hard constraint) with the project group then are randomly placed in the chromosome array (No Duplicates!)

FITNESS FUNCTION:

Conflicts are calculated as follows:

Using the collections.Counter module, we analyze if an examiner/supervisor is repeated in any of the parallel sessions (inner array) if such exists we increment conflicts by the number of repeated values

Using the same module we flatten then analyze the whole chromosome in order to count the number of projects assigned to each instructor to optimize it between 3 and 6 and balancing the number of projects assigned between instructions, conflicts incremented as such mentioned.

Moving a window of 3 over the inner arrays, then finding the UNION of examiners of each inner array set, if such exists meaning an examiner has three sessions in a row! Conflicts++

Solution is found when conflicts reach 0!

*Note: if we reach a local maximum we do a random restart by counting generations, for example if we reach 600 generations (20*30 ,--- 20 being population) a restart is initiated.

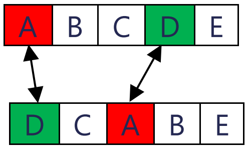
SELECTION:

Least 60% are discarded and the top 40% fittest are picked for breeding the discarded 60%

CROSSOVER:

two random chromosomes are repeatedly chosen and used to crossover and added to the population until we fulfill the missing 60%

crossover is done by choosing two random but distinct genes (random positions) and are then interchanged between both the chromosomes, for example:



MUTATION:

A gene is mutated by randomly picking another set of examiners

Mutation chance is rather high at first (10%) but then is reduced to 2% when the conflicts are reduced, which has shown a decent increase is solution speed after setting the chance to be variable!

Extra:

Data is read/written automatically using xlrd/xlwt which is a python module that handles CSV format

I managed to find previous data in ritaj from past years, copy pasted it into excel and had python do the job of pulling the data and writing a new output file of the found solution