Final Proposal

Fernandez, Ryan Austin
Poblete, Clarisse Felicia M.

Problem Space

Problem Space

- Activity Scheduling for Orgs

Inputs

- Projects
  - Project Name
  - Project Length
  - Project Date Range (what dates can it be on?)
  - Project Time Range (what times can it be on?)
  - Target Groups
  - Venue

Inputs

- Target Groups
  - Name of Group (e.g. Freshmen, ST Students)
- Restrictions
  - Blacked out days
  - Blacked out time periods
  - Venue Availability

Data Representation & Outputs

- Data Representation
  - Each activity is simply an object in an array
  - The array with complete assigned dates and times to each activity is a schedule
    - In GA, each schedule is considered a chromosome
- Outputs
  - Each project with date and time
  - Possible visualization?

Algorithms, Data Structures, Computational Models

Simulated Annealing

1. Initialize temp to 100. Start with random schedule.
2. If temp == 0 or punishment of current == 0, return current schedule
3. Else
    a. Randomly change the schedule
    b. deltaP = punishment of old sched - punishment of new sched (same as GA)
    c. If deltaP > 0, set new sched as current sched //if new sched is better, accept
    d. Else if random(0,1) < e^(deltaP/temp)
        i. Set new sched as current
        ii. Lower temperature
4. Repeat 2 - 4 until step 3 returns sched

Genetic Algorithm (Corne, et al., n.d.)

1. Generate k random schedules in population p
    a. Each schedule fits in the restrictions
2. For each schedule s in p
    a. Determine “punishment” of schedule <tentative heuristics>
        i. For each date and time conflict, add 20 points
        ii. For each conflict in similar target group, add 70 points
        iii. For each venue conflict, add 100 points
        iv. For each activity that takes place right after another activity for the same target group, add 30 points
    b. Fitness is 1/(1 + punishment)

Genetic Algorithm

c. Normalize fitness to the interval [1,10] by
    i. newfitness = 9(currfitness-minx)/(maxx - minx) + 1
d. Assign probability to each schedule based on fitness
    i. probx = fitnessx/total fitness of p

3. Generate new population
    c. Select 2 random members of the population based on probability in 2b.
    d. Make child by crossing over the schedules of half the activities.
    e. 1% of the time, mutate the child by shifting an activity to a random schedule within its restrictions
    f. Repeat b-d k times to generate new population p
4. Repeat 2 - 3 for population p for m iterations and choose best schedule in p (if a current sched has a fitness of 1, accept)

Scope and Architecture

Scope and Limitations

- Scope
  - Only evaluate performance of genetic algorithm and simulated annealing
  - Based on requirements when scheduling activities of student organizations specifically
- Limitations
  - Only single venue, single time span and single day activities
  - Blackout times are universal for all activities

Screen Flow

Activity Scheduler

Activities | Target Groups | Settings | Generate Schedule

Activities

Add Activity | Delete Activities

Activity Title	Length	Venue	Target Groups
Something Seminar	1 hour 30 minutes	ISR	1st Year CS, 1st Year INSYS, 1st Year IT
Whatever Workshop	2 hours 30 minutes	G301	2nd Year CS-ST
Coding Competition	4 hours	Gokongwei Lobby	All

Activity Scheduler

Activities | Target Groups | Settings | Generate Schedule

Add Activity

Project Name:

Project Length: 0 hours 0 minutes

Possible Dates: Any: Monday Tuesday Wednesday Thursday Friday Saturday

Specific Dates: July 5, 2016

Possible Times: 9:00AM to 3:00PM

Target Groups: 1st Year CS

Venue:

Add Activity

Activity Scheduler

Target Groups

Add Target Group | Delete Target Groups

Target Groups
1st Year CS
1st Year INSYS
1st Year IT
2nd Year CS-CSE
2nd Year CS-IST
2nd Year CS-NE

Activity Scheduler

Activities | Target Groups | Settings | Generate Schedule

Settings

Start and End Date: July 5, 2016 to August 5, 2016

Restricticted Activity Dates:

Any: ☐ Monday ☐ Tuesday ☐ Wednesday ☐ Thursday

☐ Friday ☐ Saturday ■ Sunday

Specific Dates: July 5, 2016 ⊕

Apply Settings

Activity Scheduler

Activities | Target Groups | Settings | Generate Schedule

Generated Schedule

Activity Title	Length	Venue	Target Groups	Schedule
Something Seminar	1 hour 30 minutes	ISR	1st Year CS, 1st Year INSYS, 1st Year IT	July 8, 2016 (9AM-10:30AM)
Whatever Workshop	2 hours 30 minutes	G301	2nd Year CS-ST	July 15, 2016 (1PM-3:30PM)
Coding Competition	4 hours	Gokongwei Lobby	All	July 22, 2016 (1PM-5PM)

References

Corne, D., Fang, H., Mellish, C. (n.d.). Solving the Modular Exam Scheduling Problem with Genetic Algorithms. DAI Research Paper. Retrieved from https://www.macs.hw.ac.uk/~dwcorne/cfm.pdf

Demuelemeester, E. & Herroelen, W. (1992). A Branch-and-Bound Procedure for the Generalized Resource-Constrained Project Scheduling Problem.

Gaspero, L.D. (2003). Local Search Techniques for Scheduling Problems: Algorithms and Software Tools. Universita degli Studi di Udine, Retrieved from http://www.diegm.uniud.it/digaspero/papers /DiGasperoPhDThesis.pdf

Jankovic, M. (2008). Making a Class Schedule Using a Genetic Algorithm. Retrieved July 4, 2016, from Code Project: http://www.codeproject.com/Articles/23111/Making-a-Class-Schedule-Using-a-Genetic-Algorithm

Refanadis, I. & Yorke-Smith, N. (2009). On Scheduling Events ad Tasks by an Intelligent Calendar Assistant. University of Macedonia. Retreived from http://users.dsic.upv.es/~msalido/workshop-icaps09/papers/paper6-pag43-52.pdf

Russell, S. & Norvig, P. (2010). Artificial Intelligence: A Modern Approach, 3rd Ed. New Jersey: Prentice-Hall.

Scheduling Algorithm. (n.d.). Retrieved July 4, 2016, from Carnegie Mellon School of Computer Science: http://www.cs.cmu.edu/~gregjor/project/docs/modules/optimizer/algorithm.html