The exam-scheduling project

TimetablerEASY Trade Fair Presentation

Agenda

- Problem
- Application
- Simulated Annealing
- Live demo

The problem

Formally, the exam scheduling problem can be described as follows:

- there are N students
- there are M exams to schedule into separate time-slots
- there are S time-slots
- every student takes “L(k)” exams (“k” denotes a specific student from among all students and “L(k)” denotes a student-specific list of exams)

The problem

The problem is to schedule M exams into S time-slots so as to enable all students to take their exams.

The overall quality

With a number of time-slots that is greater than the minimum required for the solution of the exam-scheduling problem one can strive to refine the allocation of exams to time-slots in a way that creates greatest gaps between exams for individual students.

The overall quality

For students sitting two exams s time-slots apart, the approximate costs are ws i.e. w0=16, w1=8, w2=4, w3=2 and w4=1. The overall quality of the exam schedule can be measured as a sum of the costs ws averaged for all students.

The overall quality - example

For example if a student had exams scheduled in time-slots: 1, 2, 4, 8, the cost evaluated for this student’s exam schedule would be 16+8+2=26.

THIS PROBLEM IS NP-COMPLETE!

But there is a solution

TimetablerEASY

Our Application

Our Results

Our application have best result of 150 penalty points for 14 time-slots.

Other applications

Other applications have their best results with around 160 penalty points for 13 time-slots.

Application features

Works on many devices thanks for using iOS platform.

This platform is very popular and offers much flexibility.

Application features

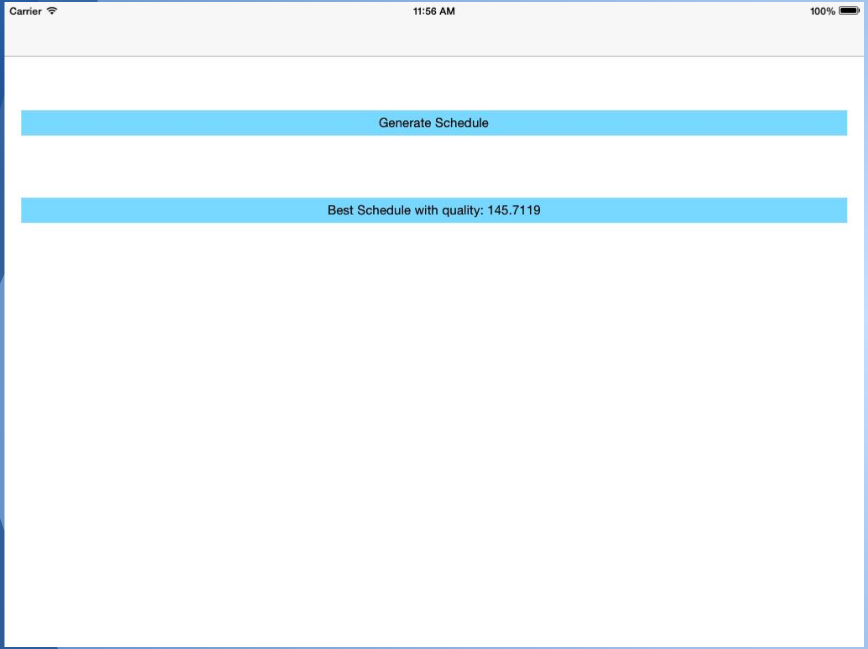
Simple interface that is easy to use and read.

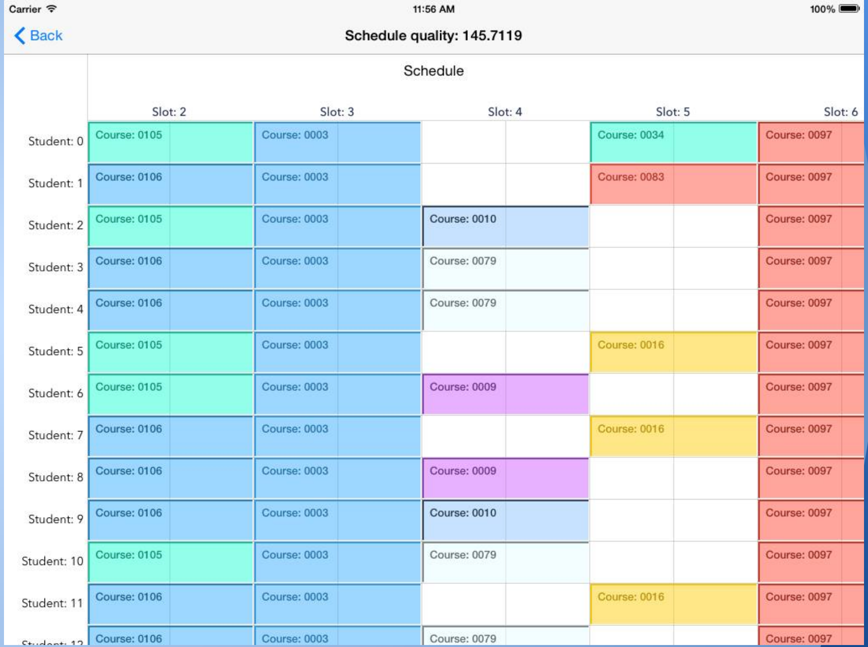
Works well with mouse-keyboard interface and also with touch screen.

Application features

Supports providing schedule data from URL.

Uses Simulated Annealing algorithm.


Carrier
11:56 AM
100%
Generate Schedule
Best Schedule with quality: 145.7119


Carrier
11:56 AM
100%
Back
Schedule quality: 145.7119
Schedule
Slot: 2
Slot: 3
Slot: 4
Slot: 5
Slot: 6
Student: 0
Course: 0105
Course: 0003
Course: 0034
Course: 0097
Student: 1
Course: 0106
Course: 0003
Course: 0083
Course: 0097
Student: 2
Course: 0105
Course: 0003
Course: 0010
Course: 0097
Student: 3
Course: 0106
Course: 0003
Course: 0079
Course: 0097
Student: 4
Course: 0106
Course: 0003
Course: 0079
Course: 0097
Student: 5
Course: 0105
Course: 0003
Course: 0016
Course: 0097
Student: 6
Course: 0105
Course: 0003
Course: 0009
Course: 0097
Student: 7
Course: 0106
Course: 0003
Course: 0016
Course: 0097
Student: 8
Course: 0106
Course: 0003
Course: 0009
Course: 0097
Student: 9
Course: 0106
Course: 0003
Course: 0010
Course: 0097
Student: 10
Course: 0105
Course: 0003
Course: 0079
Course: 0097
Student: 11
Course: 0106
Course: 0003
Course: 0016
Course: 0097
Course: 0106
Course: 0003
Course: 0079
Course: 0097

Application features

Can be used in:
Schools

Universities

Anywhere where you need a roster.

Simulated Annealing

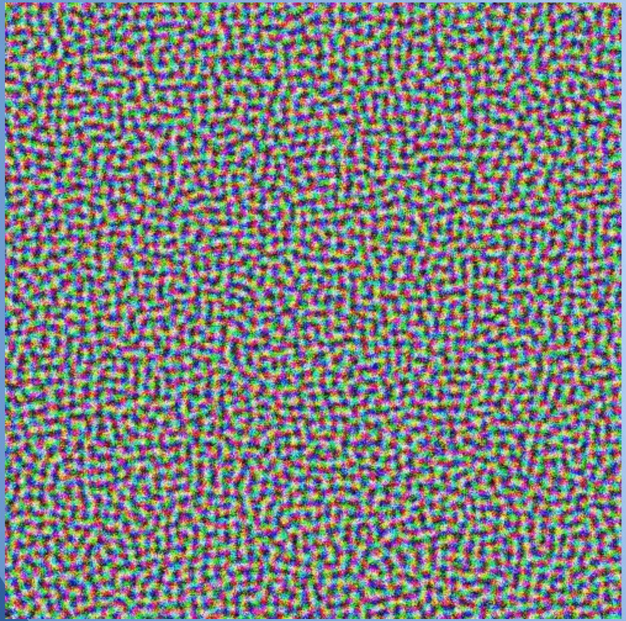
Simulated Annealing

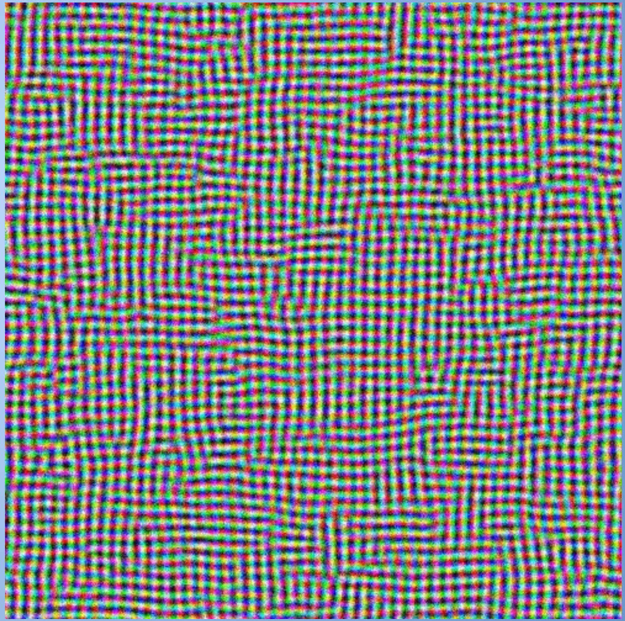
- A generic probabilistic metaheuristic algorithm
- Resolving the global optimization problem of locating a good approximation to the global optimum.
- Finds good solution in a fixed amount of time, rather than the best possible solution.

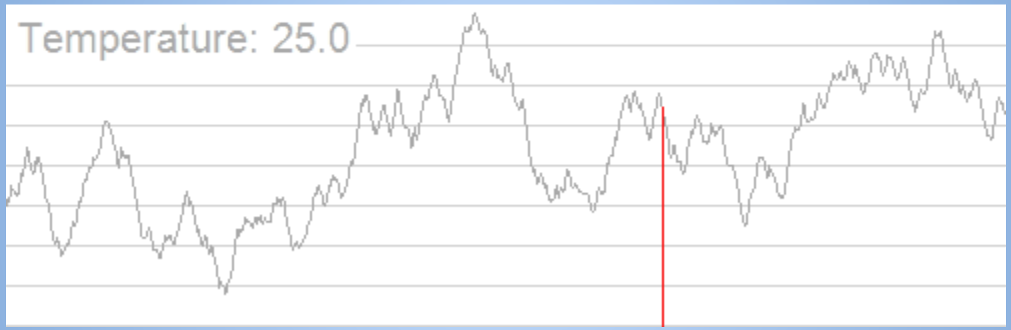
Simulated Annealing

Let s = s0
For k = 0 through kmax (exclusive):
    T ← temperature(k / kmax)
    Pick a random neighbour, snew ← neighbour(s)
    If P(E(s), E(snew), T) > random(0, 1), move to the new state:
        s ← snew
Output: the final state s

Simulated Annealing






Temperature: 25.0

LIVE DEMO