The exam-scheduling project (TimetablerEASY)

Introduction

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

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.

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.

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!

TimetablerEASY Results

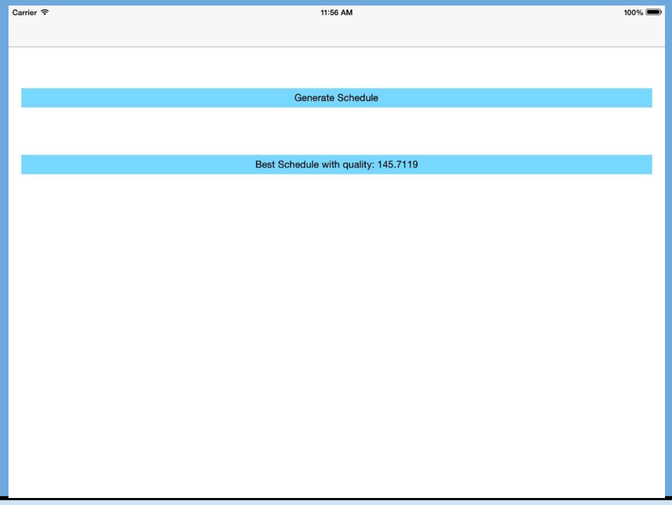
- Our application have best result of 150 penalty points for 14 time-slots.
- Other applications have their best results with around 160 penalty points for 13 time-slots.

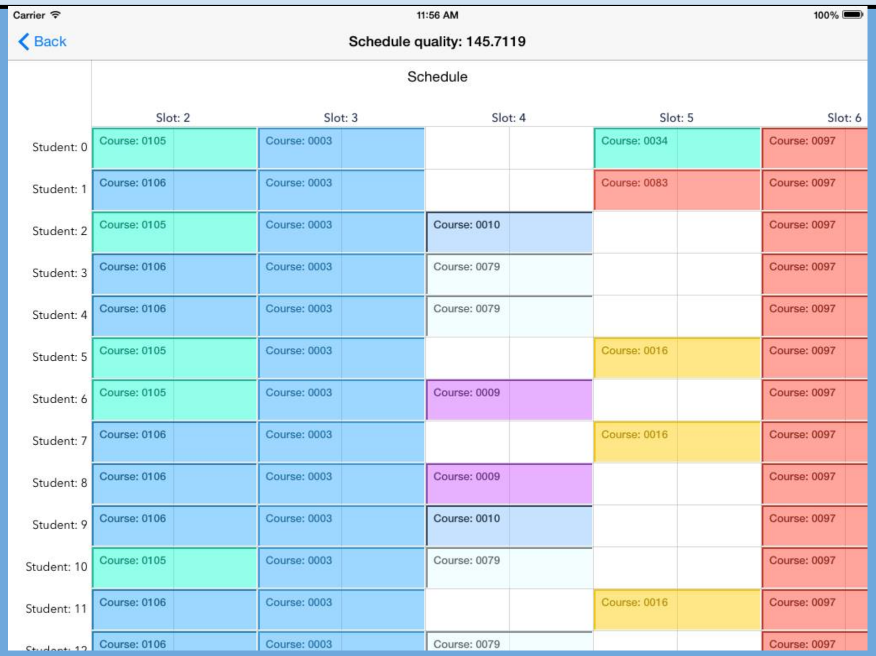
TimetablerEASY features

- Uses Simulated Annealing algorithm
- Works on many devices thanks for using iOS platform.
- This platform is very popular and offers much flexibility.
- Simple interface that is easy to use and read.
- Works well with mouse-keyboard interface and also with touch screen.
- Supports providing schedule data from URL.

Can be used in:

- Schools
- Universities
- Anywhere where you need a roster.





Application Performance

Application can be run on various platforms and its performance differs because of hardware setup.

For tests input data was set with following variables:

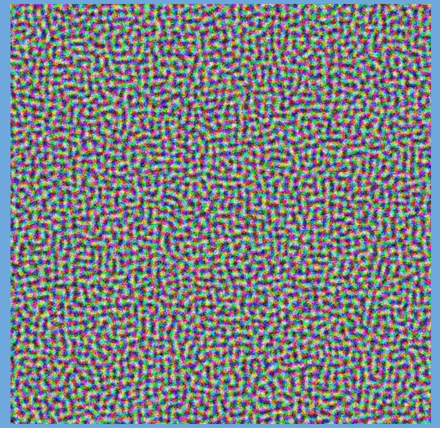
Starting temperature: 1200.95
"Frozen" temperature: 2^-6
Temperature decreases by value: 0.95
Number of students: 611
Number of exams: 113

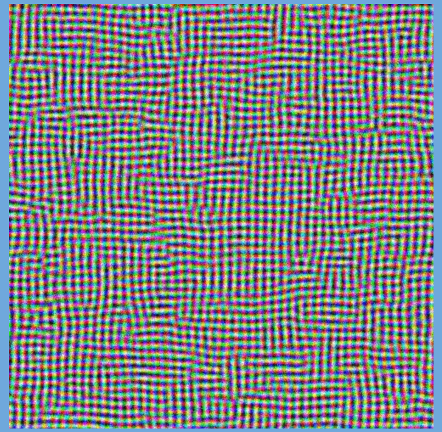
Device	Time(s)	Best result
Macbook 2010 Mid	1570	145,711
iPhone 5	3620	148,947
iPad Mini 2	670	152,189
Mac Mini 2012	420	147,685

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.

```
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
```





Example illustrating the effect of cooling schedule on the performance of simulated annealing. The problem is to rearrange the pixels of an image so as to minimize a certain potential energy function, which causes similar colours to attract at short range and repel at a slightly larger distance. The elementary moves swap two adjacent pixels. These images were obtained with a fast cooling schedule (left) and a slow cooling schedule (right), producing results similar to amorphous and crystalline solids, respectively.