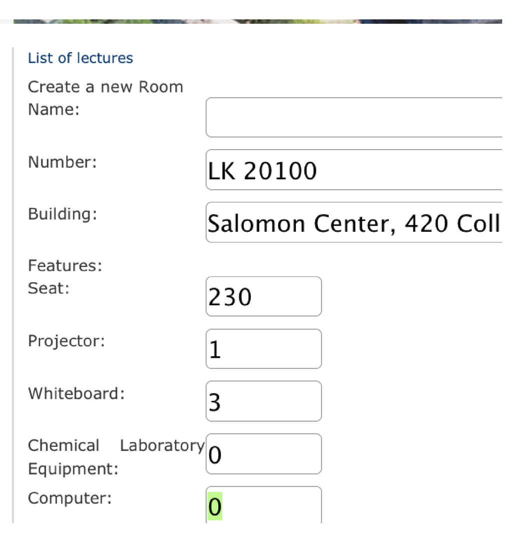
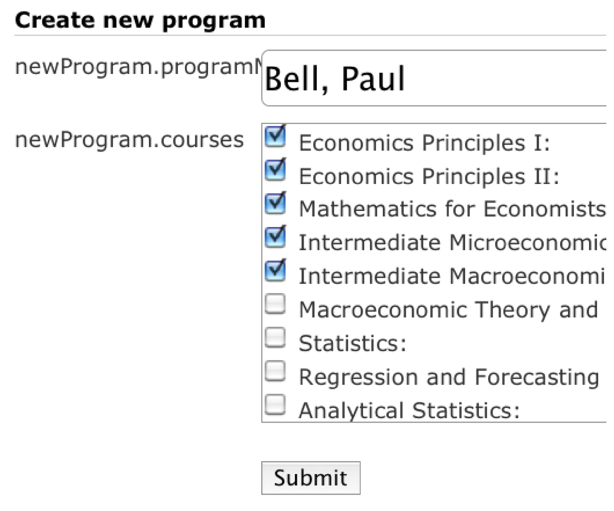
SceTris

A Course Scheduling System

Course scheduling, especially at universites and similiarly large facilities, goes along with complex constraints. With increasing input and more constraints an optimal solution cannot be computed within a reasonable amount of time. The widely used approach of using a genetic algorithm seemed to be promising and was therefore chosen for our project. The genetic algorithm operations, setup, fitness function, crossover, mutate and selection, are best conceived of as core components of the scheduler algorithm.

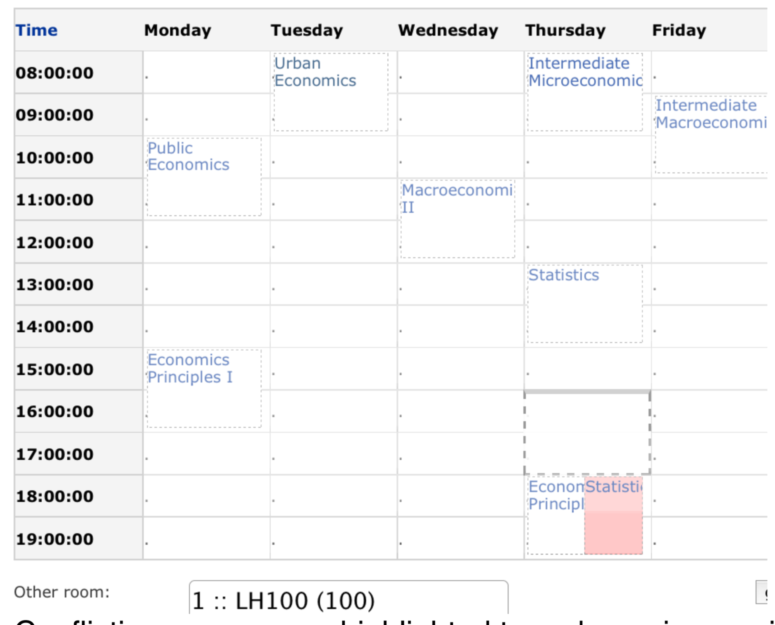
mmodo ludolutatet, commy nibh exerciduipit nis niscipit, si. Etue vedeliquis nos et inciliquat num volore conum ver summy nim nos nullum zzriustrud modo doloreet ad dipis nim zzriliquisl dolestrud doluptatem vullandre vel et prat in ulla con heniam dolor sequam, con exer sisl ute molorer summodipis nisl delissi enisc

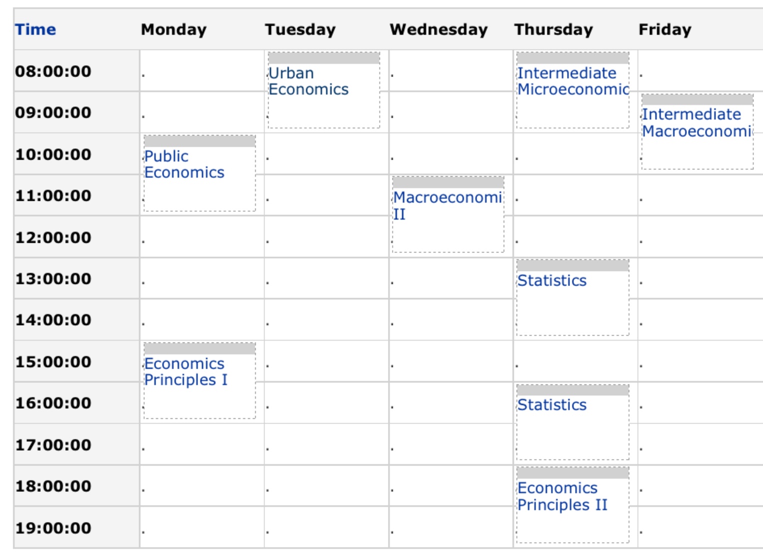




Conflicting courses are highlighted to make review easier, direct shifting of courses is possible

mmodo ludolutatet, commy nibh exerciduipit nis niscipit, si. Etue vedeliquis nos et inciliquat num volore conum ver summy nim nos nullum zzriustrud modo doloreet ad dipis nim zzriliquisl dolestrud doluptatem vullandre vel et prat in ulla con heniam dolor sequam, con exer sisl ute molorer summodipis nisl delissi enisc





Conflicting courses are highlighted to make review easier, direct shifting of courses is possible

DEFINE

REFINE

CROSS OVER

SELECT

MUTATE

Fitness function rates the candidate solution assigning it a score. The score is mapped to the number of constraints satisfied. The more constraints are satisfied the higher the candidate solution is scored. The score ranges from 0.0 to 1.0.

Crossover creates a new candidate solution by mixing and matching parts of two given candidate solutions. How the mixing and matching is done depends to the representation of a candidate solution. As our representation is a mapping of courses to allocated rooms and times, these allocations are mixed and matched.

Mutation creates a new candidate solution by taking a given candidate solution and changing a specified amount of course allocations to new, randomly chosen, course allocations. Selection iterates the given candidate solutions and keeps only the μ best solutions. The solutions are selected, according to the score given by the fitness function, through droppving the rest of the candidate solutions.

Team
David Bialik
Julian Fleischer
Hagen Mahnke
Konrad Reiche
André Zoufahl