Introductie

Het tweede algoritme dat wij geïmplementeerd hebben om onze *objective value* te verlagen, is het *restart simulated annealing* algoritme voor het verdelen van lessen over het rooster (voor het verdelen van de studenten over de lessen, gebruiken we nog steeds de hillclimber).

Dit algoritme neemt een *random* rooster, waarbij alle vakken van alle studenten zijn ingedeeld in het minimumaantal lessen (rekening houdend met het maximaal aantal studenten per les) en waarbij de studenten evenwichtig verdeeld zijn over de lessen van een vak. Voor drie tussensloten worden honderd maluspunten gerekend.

Het algoritme doet steeds een kleine *random* aanpassing. Daarna worden de nieuwe maluspunten berekend. We beginnen met een starttemperatuur, en die wordt tijdens het itereren telkens verlaagd. Op basis van de temperatuur wordt een kans berekend met de volgende formule

$$kans = 2^{\textit{(oude maluspunten – nieuwe maluspunten) / temperatuur}}$$

Als de kans hoger is dan een random getal tussen 0 en 1, wordt de wijziging doorgevoerd. Dit heeft tot gevolg dat verbeteringen altijd worden geaccepteerd, maar soms worden ook verslechteringen geaccepteerd. In het begin is de kans daarop groter, en die kans wordt steeds kleiner. Het doel hiervan is om te voorkomen dat het algoritme blijft hangen in een lokaal optimum.

Resultaten

Hieronder de resultaten van de verschillende runs. We draaien de algoritmes voor het verdelen van de lessen, en daarna het verdelen van de studenten over de lessen, 10 keer per run, en berekenen daarna het gemiddelde en het beste resultaat.
We hebben voor al deze runs dezelfde parameters voor de student hillclimber gebruikt. We hebben hiervoor de parameters gekozen die in combinatie met een hillclimber voor lessen de beste resultaten leken te geven.
We zien dat een hoger aantal iteraties leidt tot een beter resultaat. Dit lijkt logisch, maar bij de hillclimber zagen we dat (voor de eindscore) niet terug. Het leidt echter ook tot langere runtimes.
We zien geen structure invloed van de temperatuur op het resultaat.

Iteraties	*starttemperatuur*	*runtime*	*Gemiddelde*	*Minimum*	*Tussentijds Minimum (vóór herverdelen studenten)*
30000	2	169m	306	283	500
30000	4	163m	312	292	522
1000		91m	318	292	530
30000	0,5	155m	312	294	495
500		146m	336	295	567
10000	0,5	124m	328	305	549
10000	4	130m	333	307	576
10000	2	126m	332	311	533
10000	2	126m	332	311	533

Tabel 1. Hillclimber (zwart) en simaluted annealing (groen) algoritme met verschillende aantal iteraties.
Gesorteerd op minimumaantal maluspunten.