

## Introductie

Het eerste algoritme dat wij geïmplementeerd hebben om onze *objective value* te verlagen, is het *restart hillclimber* algoritme. Dit algoritme neemt een *random* oplossing van het probleem en doet steeds een kleine *random* aanpassing. Als deze aanpassing leidt tot een verslechtering van de resultaten, wordt de aanpassing teruggedraaid. Als de aanpassing leidt tot een lagere of onveranderde *objective value*, doet hij een volgende aanpassing. Het algoritme blijft aanpassingen doen tot de *objective value* N keer niet verbeterd is.

In ons geval beginnen we dus met een *random* gegenereerd 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. Dit resulteert niet per se in een geldig rooster, aangezien studenten drie tussensloten kunnen hebben, maar hiervoor worden honderd maluspunten gerekend. Aanpassingen aan dit rooster worden geaccepteerd tenzij ze in een hoger aantal maluspunten resulteren dan het vorige rooster. Bij deze aanpassingen maken we onderscheid tussen het omwisselen van lessen in het rooster en het omwisselen van studenten tussen werkgroepen en practica van hetzelfde vak.

Momenteel starten we met het omwisselen van de lessen en daarna verwisselen we de studenten tussen de werkgroepen en practica, maar hiermee willen we in de toekomst nog spelen door het bijvoorbeeld om en om te doen.

## Herverdelen van lessen

Een parameter die bij dit algoritme een rol speelt is N, ofwel het aantal iteraties dat het aantal maluspunten maximaal gelijk mag blijven of mag verslechteren voordat het algoritme stopt.

## Herverdelen van studenten

Binnen het onderdeel waarin we de individuele studenten tussen les van eenzelfde vak verwisselen, worden steeds alle zelfde soort lessen van een *random* vak geselecteerd en dit wordt herhaald tot het aantal maluspunten L keer niet is verbeterd. Iedere keer dat de lessen geselecteerd zijn, worden de studenten hierbinnen omgewisseld tot het aantal maluspunten S keer is verslechterd of gelijk is gebleven.

## Resultaten

De verschillende parameters resulteren niet in drastisch uiteenlopende *objective values*, maar er vallen wel wat zaken op. Zo lijkt een hogere S gunstig te zijn voor zowel de gemiddelde als de minimum *objective value*. Ook lijkt een hogere L tot een lagere minimum *objective value* te leiden, alhoewel niet per se tot lager gemiddelde. N lijkt in dit alles niet een erg grote rol te spelen. De configuratie die de laagste gemiddelde *objective value* oplevert is er een met N=1000, L=500 en S=50.

N	L	S	Gemiddelde	Minimum	Tussentijds Minimum (vóór herverdelen studenten)
500	300	50	326	273	541
1000	500	50	318	292	530
500	500	50	336	295	567
1000	500	20	326	302	531
1000	300	50	332	311	525
500	500	20	349	311	562
500	300	20	338	312	543
1000	300	20	336	313	527

Tabel 1. Hillclimber algoritme met verschillende parameters N, L en S. Gesorteerd op minimumaantal maluspunten.