

Algoritme 5 punten:

- We hebben geprobeerd om de simulated annealing te optimaliseren, door in plaats van huizen te verplaatsen, deze te switchen. Maar dit blijkt, zoals te zien in de onderstaande experimenten geen verbetering op te leveren (experiment 1). Dit extra algoritme is te vinden in het bestand *alg_simannealing_switch.py* in de map *Functions*. Echter blijkt dat de combinatie met de hill climber wel voor betere scores zorgt (zie experiment 3).

Experimenten. 5 punten:

1. Simulated Annealing met directions vs Simulated Annealing met switchen

In dit eerste experiment checken we in hoeverre twee varianten van simulated annealing effect hebben op de score. In de eerste variant (directions) werkt het algoritme als volgt:

1. Kies een random huis
2. Kies een random kant
3. Verplaats het huis richting deze kant
4. Neem verbeteringen aan, bij verslechtering, bereken de kans van toelaten aan de hand van de temperatuursfunctie: $e^{((\text{nieuwe score} - \text{oude score}) / \text{temperatuur})}$. Waarbij verslechtingen worden aangenomen als de uitkomst van deze functie groter is dan een random getal tussen 0 en 1.
5. Anders verwijder de verslechtering
6. Algoritme herhaalt stap 1 - 5 totdat de temperatuur kleiner over gelijk is aan 1.

De tweede variant (switchen) werkt vrijwel op dezelfde wijze, alleen stap 1 en 2 zijn anders:

1. Kies twee willekeurige huizen (die niet van hetzelfde type zijn)
2. Wissel de coördinaten om.

Voor elke variant hebben we deze battle bekeken en de volgende score resultaten zijn hier uitgekomen:

Algoritme	Type 20	Type 40	Type 60
random	11.365.950	17.858.670	24.239.310
directions	11.500.590	17.984.430	24.433.800
switchen	11.344.410	17.872.770	24.417.750

Ook na nog een extra run ter bevestiging blijkt dat switchen minder presteert:

switchen	11.339.550	17.780.550	24.152.220
----------	------------	------------	------------

Wanneer we de twee vergelijken is er geen significant verschil tussen de scores, ondanks dat de 'directions' variant vaak meer scoort.

Conclusie

De "switch" variant verschilt ondanks dat hij lager scoort niet significant van de 'directions' variant en kan daarom niet beter worden genoemd,

2. Hill Climbing algoritme → Simulated annealing (met wisselen van huizen) of andersom.

In dit experiment willen we testen in welke volgorde we de algoritmes het beste kunnen uitvoeren. Belangrijk hierbij te vermelden is dat onze simulated annealing geen variant is op onze hill climber. Onze hill climber pakt namelijk alle richtingen en verplaatst de huizen net zo lang totdat de score daalt, terwijl onze simulated annealing in deze test enkel de huizen switched van positie op de kaart. Beide sessies gebruiken de random kaart als startpunt. Na het runnen kwamen de volgende resultaten naar voren (HC = hill climber, SASWITCH = simulated annealing (switch)):

Algoritme	Type 20	Type 40	Type 60
Random	11.365.950	17.858.670	24.239.310
HC → SASWITCH	12.602.910	20.018.670	25.674.120
SASWITCH → HC	13.565.370	20.299.770	26.676.090

Bij type 20 lijkt er een significant verschil te zijn, waarbij de prijs 7,6% hoger wordt door eerste simulated annealing uit te voeren en daarna de hill climb.

Bij type 40 is er geen significant verschil, maar wel een prijsverbetering van 1,4%

Bij type 60 is er ook geen significant verschil, maar wel een prijsverbetering van 3,9%.

Conclusie

Bij het combineren van de algoritmen lijkt het beter om eerst simulated annealing met switch uit te voeren voordat onze hill climber wordt uitgevoerd.

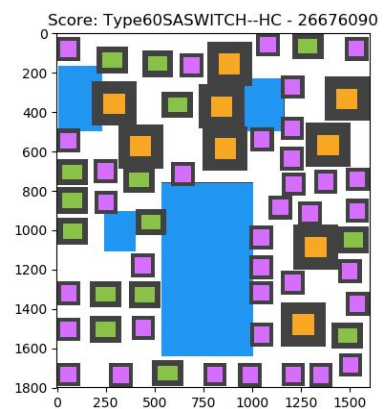
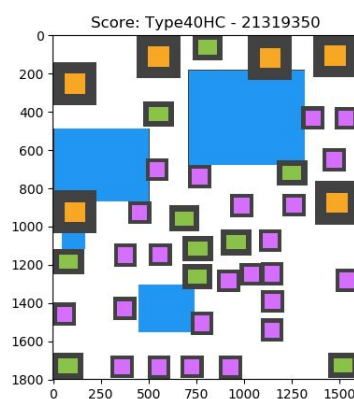
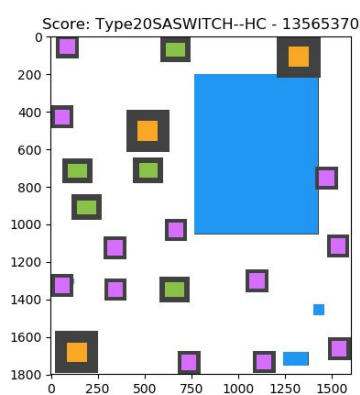
3. Welke van de algoritme scoort het beste?

Ten slotte testen we welk algoritme het beste is. Is het, het random algoritme, simulated annealing, hill climber of de ultieme combinatie? De tabel laat de resultaten zien:

Algoritme	Type 20	Type 40	Type 60
Random	11.365.950	17.858.670	24.239.310

Random + Hill climber	12.833.490	21.319.350	25.922.070
Random + SA	11.500.590	17.984.430	24.438.800
Random + SASWITCH	11.344.410	17.872.770	24.417.750
Random + HC → SASWITCH	12.602.910	20.018.670	25.674.120
Random + SASWITCH → HC	13.565.370	20.299.770	26.676.090

Zowel type 20 als 60 hebben de beste scores via de combinatie van alle drie de algoritmes. Alleen type 40 heeft geen voordeel gehad aan de combinatie en blijkt door de hill climber de beste scores te hebben verkregen. Hier de afbeeldingen van de beste resultaten:



Conclusie

Is ons geval is het het beste om alle algoritme met elkaar te combineren in de volgorde Random → Simulated Annealing (switch) → Hill climber. Waarbij rekening moet worden gehouden dat de hill climber alleen, ook een goed score kan opleveren.