



00 INTRODUCTIE Dit verslag beschrijft de zoektocht naar een heuristiek om plattegronden voor de fictieve woonwijk Amstelhaege te ontwerpen waar de waarde van die wijk geoptimaliseerd is. Deze woonwijk moet bestaan uit 20, 40 of 60 huizen en voor 20% uit water. De huizen bestaan vervolgens weer uit 3 typen, met elk andere eigenschappen. De indelingsmogelijkheden van de wijk zijn onmetelijk. Het benaderen van een waardevolle plattegrond is dan ook een probleem waarbij een slimme heuristiek uitkomst kan bieden.

- ☐ Huizen kunnen niet over elkaar of over water geplaatst worden
- ☐ De afmetingen van de huizen staan vast (maison = 11x10,5m; bungalow = 10x7,5m; ééngezinswoning = 8x8m)
- ☐ 20% van de plattegrond moet water zijn
- ☐ De verhouding van het voorkomen van de huizen staat vast (15% maison, 25% bungalow, 60% ééngezinswoning)
- ☐ Een maison heeft 6 meter vrije ruimte rond de muren nodig, een bungalow 3 meter en een eengezinswoning 2 meter. Dit is gemeten tot de muren van het dichtstbijzijnde huis of de rand van de wijk)

02 THEORIE

Deze case is een typisch optimalisatie probleem. Een oplossing is snel gevonden, maar de *béste* oplossing is onbekend. De restricties bij het indelen van de plattegronden maken de case Amstelhaege tot een begrensd optimalisatie probleem (*constraint optimization problem*). Om goeie plattegronden te vinden is het nodig de afstand tot het dichtstbijzijnde huis te weten. Omdat de huizen geplaatst zijn in een assenstelsel, lijkt dit op het ***closest pair of points problem***. Het gaat in dit geval om een versimpelde variant, omdat één van de twee punten telkens vast staat. Namelijk het huis waarvan de vrije ruimte berekend wordt. Met ***nearest neighbour search*** [2] wordt de afstand tot het dichtstbijzijnde huis berekend. Hiervoor kunnen verschillende zoek algoritmes worden gebruikt [1]. Het meest simpel is ***linear search***. Hierbij gaat een algoritme voor elk punt op het grid na hoe dichtbij het ligt en slaat het de laagste afstand op. Op deze manier wordt van alle huizen bepaald wat als vrije ruimte rond het huis gerekend kan worden.

03 AANPAK Om meer zicht te krijgen op de complexiteit van het probleem zal eerst gekeken worden naar de toestandsruimte. Dit is de verzameling van alle mogelijke oplossingen voor het probleem, oftewel alle mogelijke plattegronden. Hoe groter deze verzameling is, hoe tijdrovender het is om met brute force (zonder algoritme of heuristiek de hele toestandsruimte af gaan) de optimale oplossingen te vinden. De bedoeling van dit project is dan ook om sneller tot een optimale oplossing te komen door het gebruik van algoritmes en heuristieken. Om een zo optimaal mogelijke plattegrond te vinden worden eerst op geldige wijze het water en de huizen op een plattegrond geplaatst. Er worden 100.000 semi-random plattegronden per aantal huizen gegenereerd. Van deze reeks random plattegronden worden de beste en slechtste plattegronden opgeslagen. Vervolgens zullen verschillende algoritmes proberen deze random, geldige plattegronden, te verbeteren. Hiervoor worden onder andere een **hill climber** en **simulated annealing** algoritme gebruikt. Zo wordt er stapsgewijs gewerkt richting de plattegrond met de hoogste waarde. Naast de random gegenereerde plattegronden worden ook hard gecode scenario's als startmap gebruikt. Dit zijn plattegronden die mogelijk interessant vergelijkingsmateriaal opleveren. Hetgeen kan resulteren in handige heuristieken die de algoritmes kunnen begeleiden in het (eerder) vinden van een betere oplossing. Scenario's zijn bijvoorbeeld alle huizen gecentreerd in de plattegrond plaatsen en alle huizen een gelijke verdeling geven. Tijdens het proces zijn visualisaties en data verzameld en daar zijn verschillende statistische test op los gelaten.

04 TOESTANDSRUIMTE De complexiteit van het probleem kan worden weergegeven aan de hand van een benadering van de toestandsruimte grootte. Dit is de omvang van de verzameling mogelijke plattegronden en deze verschilt in voor elk van de drie scenario's. Er is voor gekozen hier meer in te gaan op het aantonen dat de toestandsruimte onmenselijk groot is dan de specifieke toestandsruimte voor de drie hoeveelheden huizen op te sommen.

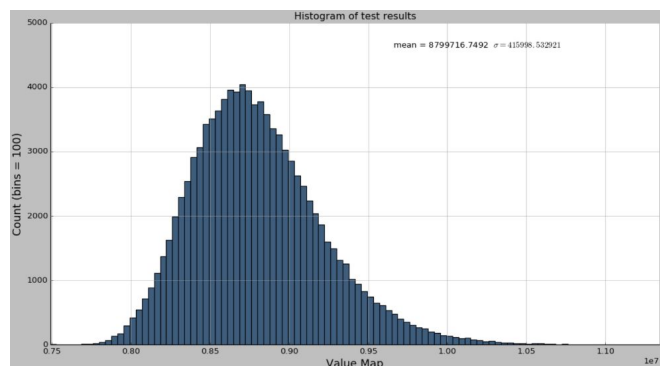
Voor 20 huizen kan de ondergrens van de toestandsruimte worden benaderd door de plattegrond in vakjes in te delen waar de verschillende huizen op geplaatst kunnen worden. Elk van deze vakjes is groot genoeg om er een huis geldig in te plaatsen (dus inclusief zijn minimaal vereiste vrije ruimte). Om de beschikbare oppervlakte te vinden voor het plaatsen van huizen verminderen we het totale oppervlakte met de 20% vereiste water. Daarna zijn de mogelijkheden om de verschillende typen huizen op de overgebleven oppervlakte te plaatsen berekend. Dit is gedaan per type, van groot naar klein, waarbij steeds de mogelijke combinaties per huis zijn berekend en vervolgens de gebruikte oppervlakte van het beschikbare oppervlakte is afgetrokken om de toestandsruimte van het volgende huistype over de overgebleven oppervlakte te berekenen. De toestandsruimte voor alle type huizen is met elkaar vermenigvuldigd. Dit komt tot een benadering van de ondergrens van: $2.44 \cdot 10^{31}$. Bij deze benadering zijn de mogelijke verdelingen die het water kan hebben nog niet mee gerekend. Dit zou de toestandsruimte nog groter maken. Omdat de ondergrens groter is dan 10^{14} is het onmogelijk om met brute force op zoek te gaan naar de beste oplossing, door alle oplossing langs te lopen.

05 HEURISTIEKEN Heuristieken zijn vuistregels die er voor zorgen dat er minder operaties nodig zijn om tot de oplossing van een probleem te komen. De eerste heuristiek die wij hebben toegepast betreft de plaatsing van het water. Bij het bepalen van de linkeronderhoekcoördinaten van het water wordt een aantal beperkingen opgelegd. Wanneer er slechts één vlak water wordt geplaatst, wordt de linkeronderhoek van dit vlak altijd ergens linksonder in de kaart geplaatst.

Zo wordt verzekerd dat het watervlak in de kaart past. Ook de andere stukken water worden niet compleet willekeurig geplaatst, maar krijgen bepaalde restricties mee: ze worden nooit heel dichtbij de rechter rand geplaatst. Dit is een heuristiek waarbij we het gevoel hebben dat de kaarten gemiddeld hogere waardes hebben dan wanneer het water compleet random geplaatst wordt. De vrije ruimte rond het water en in het midden van de plattegronden zorgt ervoor dat het water wordt benut als vrije ruimte. Dit werd bevestigd door *hardcoded* kaarten met water tegen de randen die niet meer opbrachten dan slechte random gegenereerde mappen.

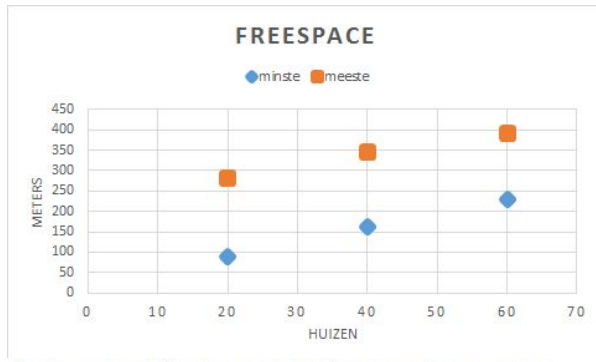
Een andere heuristiek die wordt toegepast heeft betrekking op de plaatsing van de huizen. Deze worden ook niet volledig willekeurig geplaatst. De volgorde waarin de verschillende types huizen worden geplaatst, is namelijk van tevoren vastgesteld. Dit doet niet zoveel voor de waarde van de kaarten, maar vermindert wel sterk de runtime van het programma. Door deze heuristiek lopen we echter niet de gehele toestandsruimte af bij het genereren van plattegronden.

06 SEMI RANDOM SAMPLING Om random plattegronden te genereren is een algoritme geschreven dat het water en de huizen semi-willekeurig per object een geldige positie op een vlak geeft. Er is voor gekozen eerst het water te plaatsen en dan de huizen, omdat het plaatsen van grote stukken water tussen reeds geplaatste huizen erg lastig bleek. Bij het plaatsen van het water wordt mee gegeven in hoeveel stukken dit verdeeld is (1, 2, 3 of 4) en met de tendentie de eerste stukken water meer aan de zijanten te plaatsen (ter vermindering van de kans dat het lang zoeken is naar een mogelijkheid de overgebleven stukken water te plaatsen). De ratio per water wordt wel volledig random bepaald. Na het water, worden de huizen geplaatst, in volgorde van groot naar klein. Er zijn dus bij het zogenaamde random algoritme een aantal restricties ingebouwd ter bevordering van de run-time. Hierdoor is er dus eigenlijk meer sprake van een **restricted random sampling**.

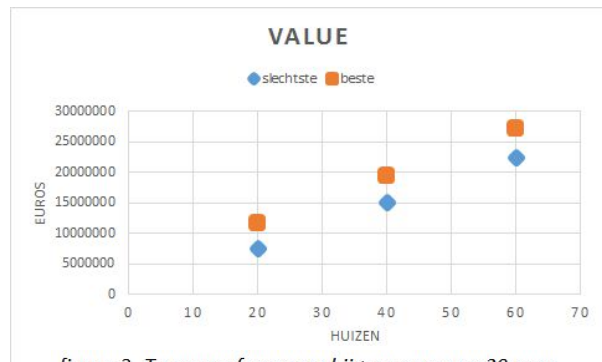


figuur 1. Histogram waarde random plattegronden met 20 huizen

06A SEMI RANDOM SAMPLING - RESULTATEN Uit de semi-willekeurig gegenereerde plattegronden komen voor de totale waarde de volgende maxima naar voren: €11611890,- (20 huizen), €19429770,- (40 huizen) en €27217110,- (60 huizen). Voor de vrije ruimte waren de maxima als volgt: 283 meter (20 huizen), 347 meter (40 huizen) en 393 meter (60 huizen). De waardes zijn normaal verdeeld, maar de verdeling is niet perfect symmetrisch (zie figuur 1). De reden hiervoor is dat ook de waardes van indelingen begrenst is aan de onderkant door de minimale map waarde, namelijk de cumulatieve waarde die de huizen hebben zonder enige *extra* vrije ruimte. De waardes lopen aan de rechterkant ver door, er is dus een grotere verdeling aan verbeteringen die op een gemiddelde map gemaakt kunnen worden. De waarde en de hoeveelheid vrije ruimte stijgt naarmate er meer huizen op de plattegrond worden geplaatst. De stijging van de vrije ruimte vliegt steeds verder af terwijl de stijging van de waarde met dezelfde grootte blijft toenemen (zoals te zien in figuur 2 en 3 op de volgende pagina).



figuur 2. Waardevermeerdering bij toename van 20 naar 40 naar 60 huizen.



figuur 3. Toename freespace bij toename van 20 naar 40 naar 60 huizen.

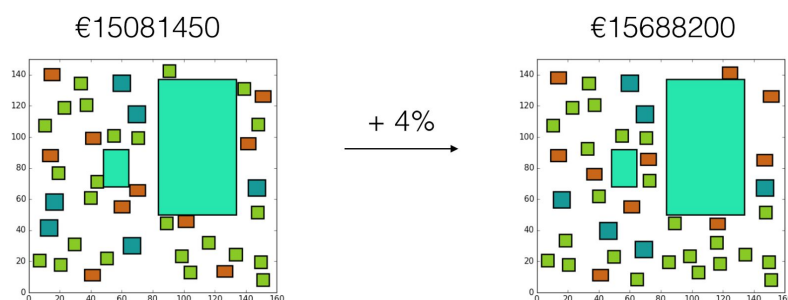
07 HILL CLIMBER Een *Hill Climber* is een algoritme dat iteratief probeert een kleine mutatie te maken en deze alleen daadwerkelijk doorvoert wanneer dit leidt tot een waardevermeerdering. Er wordt dus over de huizen op een gegeven startmap geïtereerd en steeds per huis getracht deze een nieuwe geldige positie op de map te geven. De mutatie is ofwel een random nieuwe positie op de kaart, ofwel een verplaatsing van een huis met max. 1 meter ten opzichte van zijn huidige plaats.

07A HILL CLIMBER - RESULTATEN Het *Hill Climber* algoritme is toegepast op de slechtste en op de beste kaarten. Opvallend is dat de *Hill Climber* bij de beste kaart een veel kleinere verbetering kent dan bij de slechtste kaarten. Wij denken dat dit komt doordat de beste kaarten al erg goed zijn en de *Hill Climber* snel op een lokaal optimum terecht komt. Ook blijkt dat spreiding van de verbetering in waarde door de *Hill Climber* groter is bij relatief slechte startmappen. Deze mate van spreiding neemt af met de hoeveelheid huizen op de startmap.

Hill Climber	financial*	
huizen	slechtste	beste
20	10.46 - 20.49 %	0 - 1.65 %
40	0.35 - 1.99 %	0 - 4.02 %
60	0.10 - 0.71 %	0 - 0,16 %
* spreiding verbetering tov Random Sampling		

* spreiding verbetering tov Random Sampling

tabel 1. Procentuele waardevermeerdering door de Hill Climber



07 WATER - RESULTATEN Er is gekeken naar het verschil in waarde bij kaarten met 1, 2, 3 of 4 stukken water. In de random gegenereerde kaarten is er geen verschil in waarde tussen deze scenario's. Na het runnen van de *Hill Climber* over de plattegronden blijken ook nu de verschillen niet groot, maar lijken bij 20 huizen de plattegronden met 3 of 4 stukken water iets meer te kunnen verbeteren. In tabel 2 is te zien hoeveel winst de *Hill Climber* op een map behaalt door hem tien keer te testen. Bij plattegronden met 40 en 60 zijn de verschillen na de *Hill Climber* kleiner: tussen maximaal 1.99 % en 4.02% verbetering bij 40 huizen en tussen

maximale winst van 0% tot 0.98% bij 60 huizen. Hoe meer huizen er geplaatst worden in Amstelhaege, hoe minder het aantal vlakken water de opbrengst van de wijk beïnvloedt.

De maximale verbetering die de *Hill Climber* had voor de beste kaart (20 huizen) bij 1 t/m 4 stukken water is respectievelijk: 0.97%, 1.65%, 1.80%, 2.92%. Vermoedelijk komen de huizen bij 1 stuk water eerder dit grote water object tegen, waarna ze niet verder kunnen verschuiven. Bij 4 kleinere stukken water kan een huis verder in de goede richting verschuiven.

water	ranking	startmap	Δ best of hillclimber 10 times	best € after hillclimber
1	best	€11,341,710	0 - 0.97 % verbetering	€11,451,720
	worst	€7,490,670	10.46 - 20.49 % verbetering	€9,025,508
2	best	€11,611,890	0 - 1.65 % verbetering	€11,803,490
	worst	€7,653,360	9.79 - 19.71 % verbetering	€9,161,837
3	best	€11,381,910	0 - 1.80 % verbetering	€11,586,780
	worst	€7,652,220	9.12 - 27.38 % verbetering	€9,747,397
4	best	€11,410,950	0 - 2.92 % verbetering	€11,744,150
	worst	€7,661,910	2.30 - 27.74 % verbetering	€9,787,234

tabel 2. Opbrengst van semi-willekeurige plattegronden met 20 huizen vóór en na verbetering door de hillclimber

07 SIMULATED ANNEALING Waar een *Hill Climber* algoritme een (lokaal) optimum vindt, is dit optimum niet ook altijd vanzelfsprekend het globale optimum. Het kan namelijk goed zo zijn dat de startmap waar mee gewerkt wordt zodanig ingedeeld is dat een globaal optimum niet meer bereikt kan worden door alleen maar mutaties aan te nemen die direct tot een waardevermeerdering leiden. Een *Simulated Annealing* algoritme heeft meer kans richting het globale optimum te bewegen omdat het niet alleen maar mutaties toelaat die direct beter zijn, maar ook -met een afnemende kans- mutaties toelaat die slechter zijn. Door deze verslechtingen kunnen stappen in de toekomst wel tot hogere waardes leiden die in eerste instantie niet bereikt konden worden. De kans waarmee het algoritme deze verslechterende mutaties toch toelaat is afhankelijk van het verschil in waarde tussen de nieuw gegenereerde plattegrond, de vorige plattegrond, en de temperatuur. Dit laatste is een variabele die met het aantal iteraties afneemt. Het verschil in waardes tussen de plattegronden wordt meegenomen zodat grote verslechtingen met een kleinere kans worden aangenomen, en de temperatuur zodat de acceptatiekans over de iteraties afneemt. Omdat de geldwaardes van de slechtste en beste plattegronden zo sterk verschillen zou er telkens een ander *cooling scheme* nodig zijn. Omdat de kans groter is dat de beste kaart na verbetering hoog uitkomt, wordt *Simulated Annealing* alleen gerund over de beste kaarten. De acceptatiekans wordt berekend met de formule:

$$\text{accept} = e^{(value_{new} - value_{old}) * 0.0001 / temperature}$$

Wanneer deze acceptatiekans groter is dan een random gegenereerd getal tussen 0 en 1, wordt een mutatie die de waarde van de plattegrond vermindert, toch aangenomen.

07A SIMULATED ANNEALING - RESULTATEN Het *Simulated Annealing* algoritme was lastig om geslaagd te implementeren. Dit mede omdat de verschillen tussen de waarde van mappen een grote spreiding besloeg. Hierdoor klapte de acceptatie kans vaak te snel naar nul. Over de iteraties heen sloeg ook meestal de acceptatiekans te abrupt naar nul. Er is voor gekozen het algoritme zo te bouwen dat het vooral op al redelijk goede (financiële) mappen redelijk werkt. Echter is het niet gelukt dit zo te bouwen dat het hogere mapwaarden bereikt dan ons *Hill*

Climber algoritme. Wel is duidelijker te zien dat de maisons meer naar elkaar toetrekken, omdat dit algoritme de kaarten langer blijft veranderen.

11 CONCLUSIES/REFLECTIE In deze opdracht is getracht om een optimale indeling van Amstelhaege te ontwikkelen. Om dit te doen is de wijk eerst semi-willekeurig ingedeeld en is deze indeling verbeterd door middel van optimalisatiealgoritmes. Hiermee is een indeling gevonden die een stuk meer financiële waarde en vrije ruimte oplevert dan de gemiddelde map. Verder hebben we door het bestuderen van de resultaten en plattegronden vermoedens ontwikkeld die het waard zijn verder te onderzoeken.

Er is op de beste plattegronden tot nu toe nog ruimte voor verbetering. We vermoeden dat de optimale waarde nog een stuk hoger ligt dan de hoogste waarde die wij gevonden hebben. Zelfs bij goede mappen, is met het blote oog soms te zien hoe deze nog verbeterd zouden kunnen worden. Onze verbeteringsalgoritmes moeten verder worden geoptimaliseerd om die waardes te vinden. Het *simulated annealing* algoritme moet bijvoorbeeld beter geconfigureerd worden. Een correct *cooling scheme* zou ertoe moeten leiden dat het *simulated annealing*-algoritme tot een hoger optimum komt in plaats van plotseling bevroest. Nu vinden de optimalisatiealgoritmes een lokaal optimum, in plaats van het globale optimum.

Een andere manier om een hogere waarde te bereiken is door meer heuristieken toe te passen. Vermoedelijk is er een bepaalde indeling van het water die vaak tot waardevolle plattegronden leidt en die de verbeteringsalgoritmes effectiever maakt. Als er nog meer tijd was voor dit onderzoek, zouden we via statistische analyse naar de optimale plaatsing van het water kunnen zoeken. Uit de goede plattegronden blijkt dat de maisons relatief meer vrije ruimte rond zich hebben dan de andere typen huizen. Deze vrije ruimte levert het meeste op. Er kan daarom als heuristiek tevens toegevoegd worden dat de maisons hun vrije ruimte zoveel mogelijk moeten delen. Op deze manier wordt er optimaal gebruik gemaakt van deze (kostbare!) vrije ruimte.

Een laatste verbeterpunt is de invloed van sociale aspecten op de waarde van de wijk. Momenteel houdt de waardeberekening alleen rekening met de vrije ruimte rond verschillende type huizen. De waarde van huizen in een echte woonwijk wordt bepaald door veel meer aspecten, zoals bereikbaarheid, gelijkheid tussen bewoners en nabijheid van voorzieningen. Dit zijn allemaal factoren waar een financiële waarde aan gehangen zou kunnen worden om zo ook het sociale aspect mee te nemen in de zoektocht naar de meest optimale Amstelhaege wijk. Interessant is om het effect van zulke parameters op de snelheid van de code te analyseren aangezien de uitkomsten realistischer zullen zijn, maar de rekenkosten vele malen hoger.

OVERZICHT VAN DE BESTE PLATTEGRONDEN:

uitkomsten	aantal huizen		
waardering	20	40	60
financial	€11,803,490	€19,457,820	€27,378,440
free space	283 m	351 m	392 m

tabel 3. Waardes van de 6 beste plattegronden voor alle aantallen huizen.

BRONNEN

[1] Arya, S. Mount, D.M.N. Netanyahu, S. Silverman, R. Wu, Y.A. (1998). *An Optimal Algorithm for Approximate Nearest Neighbour Searching in Fixed Dimensions*. JOURNAL OF THE ACM, Vol. 45, Issue 6: 891-923.

[2] Chavez, G. Navarro, R.A. Baeza-Yates, J.L. Marroquin. (2001). *Searching in metric spaces*. ACM Computing Surveys (CSUR) Vol. 33, Issue 3: 273-321.