De potentie van kunstmatige intelligentie in het ontwerpproces

De mogelijkheden van Generative Design

Precedenten zijn heel bepalend in het ontwerpproces. Doorgaans bereiken ontwerpers resultaten die efficiënt, maar voorspelbaar zijn. Wat als ontwerpers zouden beginnen met minder vooroordelen en zich meer zouden richten op het ontdekken van nieuwe onverwachte resultaten?

Generative Design (GD) is het proces van het definiëren van doelen en beperkingen en het gebruik van rekenkracht om automatisch een brede ontwerpruimte te verkennen en de beste ontwerpopties te identificeren. Van oudsher werd het voornamelijk gebruikt in de maakindustrie, om tegelijkertijd meerdere geldige oplossingen te genereren op basis van productiebeperkingen uit de praktijk en productprestatievereisten, zoals sterkte, gewicht en materialen. Dankzij Generative Design kunnen ingenieurs veel sneller ontwerpen dan ooit mogelijk was en uit veel meer productiegerede ontwerpopties kiezen.

De afgelopen jaren hebben onderzoekers van Autodesk manieren onderzocht om gebruik te maken van de kracht van Generative Design voor architectonisch en constructief ontwerp. Dit artikel beschrijft de basis van Generative Design en geeft twee voorbeelden waarin het is toegepast op gebieden die interessant zijn voor de lezers van *Cement*.

Wat is Generative Design? [2])

De 'motor' van Generative Design is het genetische algoritme (GA). Dit is een algoritme dat is ontstaan in de kunstmatige intelligentie en dat wordt gebruikt om oplossingen te vinden voor optimalisatie- en zoekproblemen. Het is een van de oudste en meest populaire evolutionaire algoritmen.

De geschiedenis van het genetische algoritme begint (zoals de meeste moderne ideeën in de informatica) met het beroemde document van Alan Turing uit 1950, 'Computing Machinery and Intelligence'. In dit artikel ontwikkelde Turing vele belangrijke vroege ideeën over kunstmatige intelligentie. In deze paper onderzoekt hij ook hoe concepten uit de natuurlijke evolutie kunnen worden gebruikt om een kunstmatig intelligente machine te ontwikkelen.

De eerste beschrijving van een daadwerkelijk algoritme voor evolutionaire berekeningen werd voorgesteld door John Holland in zijn boek 'Adaptation in Natural and Artificial Systems' [4], gepubliceerd in 1975. Hoewel sommige details

[1]) *Dit artikel is een combinatie van artikelen van Autodesk-medewerkers Danil Nagy [1], [2] (lead designer en principal research scientist bij The Living group binnen Autodesk Research) en Bill Danon [3] (director for industry communications). Rogier van Nalta is Technical Solutions Executive voor de Named Accounts in Nederland.*

[2]) *Gebaseerd op een artikel van Danil Nagy [1].*

ir. Rogier van Nalta [1])
Autodesk

van het algoritme van Holland specifiek zijn voor de vroege computersystemen die hij op dat moment gebruikte, worden de meeste basisbewerkingen die hij beschrijft nog steeds gebruikt in moderne genetische algoritmen.

Voorbeeld: "to be or not to be"

Om de kracht van GA te begrijpen, wordt een voorbeeld getoond van het opnieuw creëren van de eerste zin uit Hamlet van Shakespeare, "to be or not to be", letter voor letter met een vocabulaire van 96 mogelijke toetsenbordtekens [5] (fig. 1).

Het aantal mogelijke oplossingen om de 18 plekken met 96 tekens te vullen, is $96^{18} = 4{,}8 \times 10^{35}$. Als wordt aangenomen dat het genereren en evalueren van elke optie slechts één computercyclus kost, zou het doorlopen van alle opties op een moderne 2,6 GHz-processor ongeveer $58{,}5 \times 10^{17}$ jaar duren. Met behulp van het zeer eenvoudige GA beschreven door Shiffman [5], is het mogelijk dit probleem op te lossen in 32 seconden na slechts 38.000 mogelijke oplossingen te hebben bekeken (fig. 2). Een grote verbetering ten opzichte van onze vorige berekeningen.

Hoewel er andere algoritmen en complexere implementaties van GA's zijn die dit probleem sneller kunnen oplossen, is het feit dat een algoritme geschreven met slechts 36 regels code dit probleem kan oplossen na slechts 38.000 van de $4{,}8 \times 10^{35}$ mogelijke ontwerpen te hebben bekeken, behoorlijk indrukwekkend en een bewijs van de kracht en veelzijdigheid achter GA's meest fundamentele concepten.

Basisoperatoren

Hoewel het genetische algoritme vrij goed is in het oplossen van zeer complexe problemen, wordt het algoritme zelf aangedreven door slechts vier basisbewerkingen.

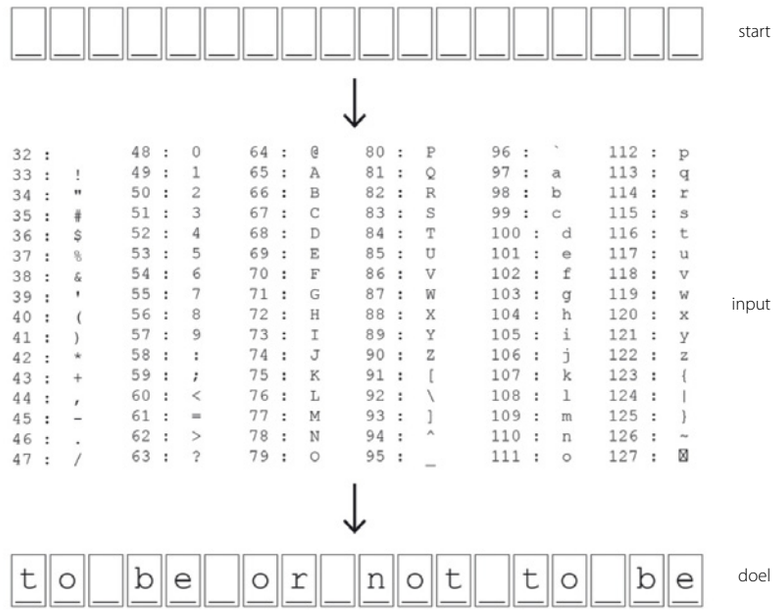
1. Generatie

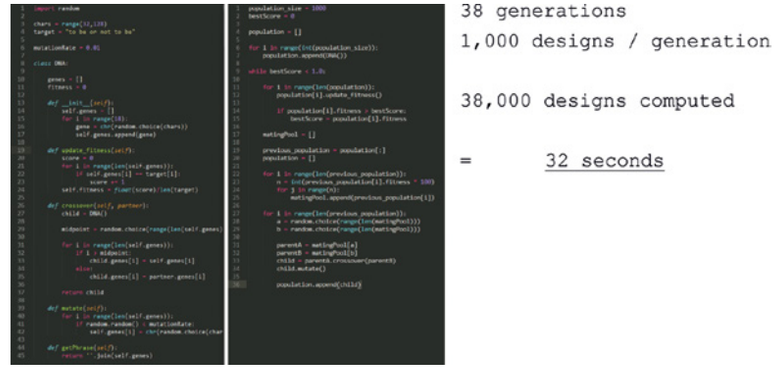
Het algoritme begint met het genereren van een reeks ontwerpen die de initiële 'generatie' vormen. GA's kunnen verschillende strategieën gebruiken voor het genereren van deze initiële ontwerpen, waaronder regelmatige bemonstering uit de ontwerpruimte met behulp van algoritmen zoals SOBOL. De meest gebruikelijke methode is echter om de ontwerpen willekeurig uit de ontwerpruimte te samplen. In het Shakespeare-voorbeeld is een populatie van 1.000 ontwerpen gebruikt.

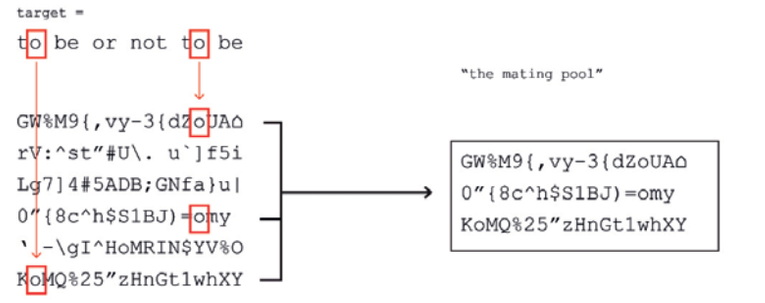
2. Selectie (ranking)

Vervolgens selecteert het algoritme welke van de eerste ontwerpen zullen worden gebruikt om de volgende generatie te genereren. Er zijn verschillende manieren om dit te doen, maar over het algemeen willen we ervoor zorgen dat betere ontwerpen een hogere kans hebben om geselecteerd te worden, zodat goede strategieën uit de eerste generatie naar de volgende worden doorgevoerd. In het voorbeeldgeval is een 'mating pool' gemaakt met de beste ontwerpen op basis van het aantal letters dat ze gemeen hebben met het doel.

1 Probleemomschrijving
2 Oplossing voor het Shakespeare-probleem met behulp van een Genetisch Algoritme
3 De selectiebewerking: links een fragment van alle ontwerpen en vergelijk met het doel, rechts een selectie van ontwerpen die (deels) voldoen aan het doel



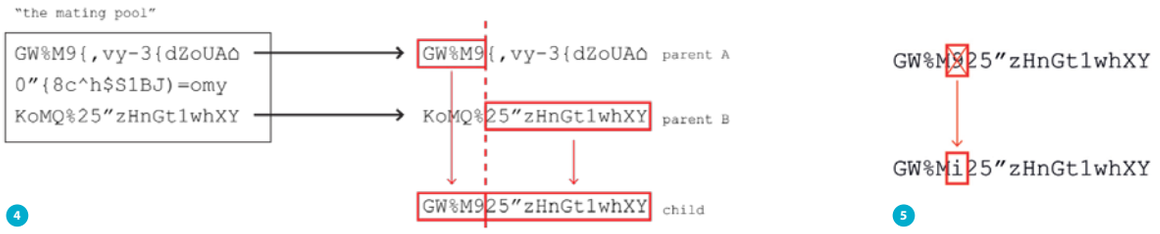




4 De cross-overbewerking: een willekeurig deel van één ontwerp wordt gecombineerd met een willekeurig deel uit een ander ontwerp. Uit de nieuwe ontwerpen worden uiteindelijk weer de beste geselecteerd voor een nieuwe cross-over

5 De mutatie bewerking; de invoer van een willekeurig aantal 'kinderen' verandert voordat ze de volgende generatie binnengaan

6 GA-proces voor het vinden van een oplossing voor het Shakespeare-probleem



4 5

3. Cross-over

Zodra het algoritme de veelbelovende ontwerpen selecteert, worden deze opnieuw gecombineerd om een nieuwe populatie van ontwerpen te maken. Dit is vergelijkbaar met het idee van 'voortplanten' in natuurlijke evolutie. Hierbij wordt genetische informatie van twee ouders willekeurig gerecombineerd om nieuw nageslacht te creëren. Het basisidee is dat, aangezien de ouders lang genoeg hebben overleefd om zich voort te planten, ze beiden een bepaald genetisch materiaal moeten hebben dat nuttig kan zijn voor het voortbestaan van de soort in het algemeen. Door elementen van de genetische informatie van zijn ouders te combineren, zal het kind waarschijnlijk een aantal winnende strategieën van beide ouders erven en dus een betere kans hebben te overleven en later te reproduceren.

De manier waarop deze cross-over wordt geïmplementeerd, is afhankelijk van de invoertypen en het specifieke GA dat wordt gebruikt. In dit geval zijn er willekeurig twee ontwerpen uit de 'mating pool' geselecteerd als 'ouders'. Vervolgens wordt een willekeurig 'cross-overpunt' gekozen en krijgt het 'kind' alle informatie tot aan het cross-overpunt van de eerste ouder en alle informatie na het cross-overpunt van de tweede ouder. Hierdoor krijgen sommige ontwerpen ('kinderen') een combinatie van eigenschappen die beter werkt dan de twee oorspronkelijke ontwerpen ('ouders'). Uit de nieuwe ontwerpen worden uiteindelijk weer de beste geselecteerd voor een nieuwe cross-over. Deze strategie wordt 'single-point cross-over' genoemd.

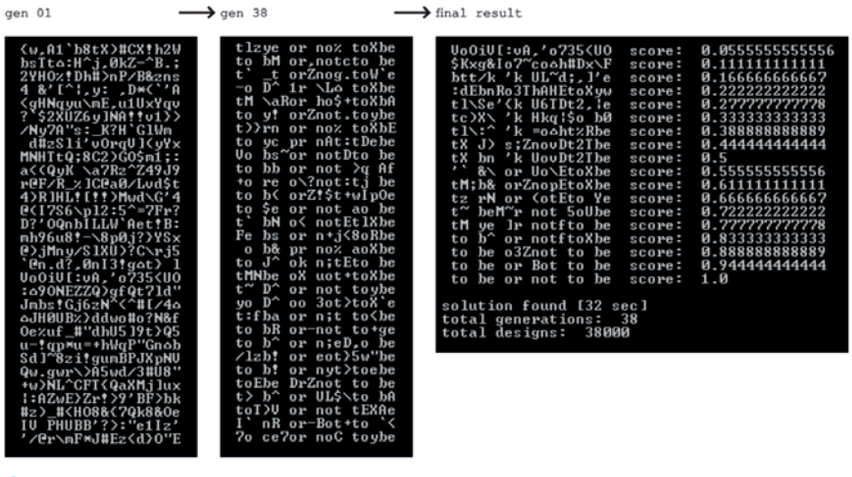
4. Mutatie

Selectie en cross-over garanderen dat de meeste van de goede strategieën van elke generatie hun weg vinden naar volgende generaties van ontwerpen. Met alleen deze methoden is het risico op vastlopen echter groot. Als bijvoorbeeld geen van de ontwerpen in de eerste generatie een 'e' had op de laatste plek, zou geen enkel kind deze informatie ooit ontvangen en zou de juiste oplossing nooit worden gevonden. Net als in de natuur is een mechanisme nodig dat nieuwe informatie willekeurig in de genenpool kan injecteren. Dit wordt gedaan door een 'mutatie'-bewerking, die willekeurig de invoer van een willekeurig aantal kinderen (meestal een klein percentage) verandert voordat ze de volgende generatie binnengaan. In dit geval wordt 1% van de input in elke generatie willekeurig opnieuw toegewezen.

Door deze bewerkingen toe te passen op een reeks generaties, kan het algoritme uiteindelijk tot de juiste oplossing komen. Figuur 6 toont een deel van de ontwerpen van de eerste en laatste generatie. Te zien is dat de ontwerpen in de eerste generatie volledig willekeurig zijn. Maar 38 generaties later hebben alle ontwerpen veel van de juiste componenten van het doel, waaronder één ontwerp dat helemaal klopt. Het rechtervenster toont het beste ontwerp in elke generatie en de score (als de verhouding van de juiste tekens). Dit laat zien hoe het algoritme in staat is de ontwerpen geleidelijk te verbeteren met elke generatie.

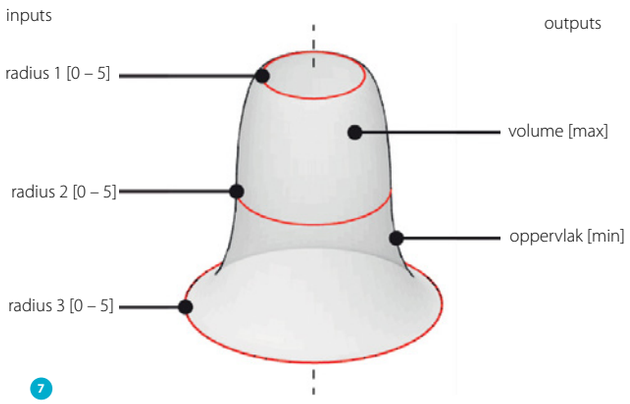
Optimalisatie op basis van meerdere doelen

Het voorgaande voorbeeld beschrijft een zeer basaal genetisch algoritme dat wordt gebruikt om een probleem met slechts één doel op te lossen: de tekenreeks zo dicht mogelijk bij het doel krijgen. De meest interessante ontwerpproblemen worden echter gedefinieerd door veel verschillende doelen, die op complexe, niet-intuïtieve manieren met elkaar in verband kunnen worden gebracht. In dit geval is het bepalen welk ontwerp beter is dan een ander niet zo duidelijk.



6

7 Model input/output
8 Visualisatie van de mogelijke ontwerpen op basis van de twee doelen



Als voorbeeld wordt een eenvoudig model beschouwd: een gesloten oppervlak dat drie cirkels met vooraf vastgestelde radius doorsnijdt. De doelen zijn het volume te maximaliseren en het oppervlak te minimaliseren. Figuur 7 toont de in- en output van het model. De input is de diameter van de drie cirkels.

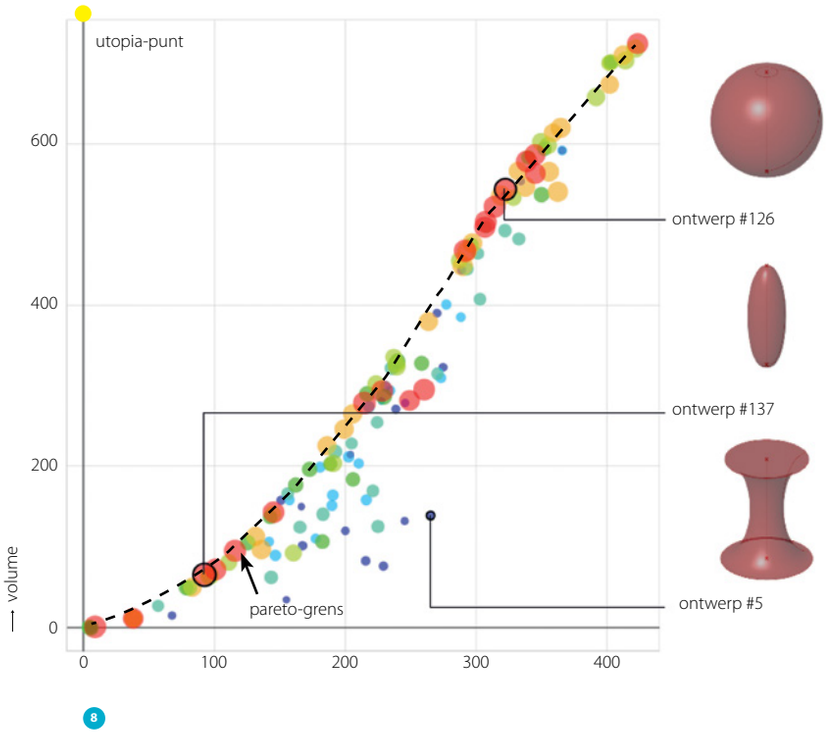
Het is gemakkelijk voor te stellen hoe deze doelen met elkaar conflicteren. Hoewel op basis van intuïtie is in te schatten dat een bol het meest efficiënte compromis van deze doelen zal geven, is er duidelijk geen enkel ontwerp dat zowel het kleinste oppervlak als het grootste volume heeft. In figuur 8 staat een reeks ontwerpen uit de ontwerpruimte die door een GA zijn gemaakt.

Figuur 8 is een 4D-scatter-plot, waarbij elke cirkel een ontwerp vertegenwoordigt dat ergens in de ontwerpruimte is gevonden. Deze grafiek is een goede manier om een ontwerpruimte te visualiseren, omdat het ons in staat stelt vier verschillende soorten informatie tegelijk te visualiseren. In dit geval staan de twee doelen langs de x- en y-as en de kleur en de grootte van de cirkels geven de volgorde weer waarin de ontwerpen door het algoritme zijn gemaakt.

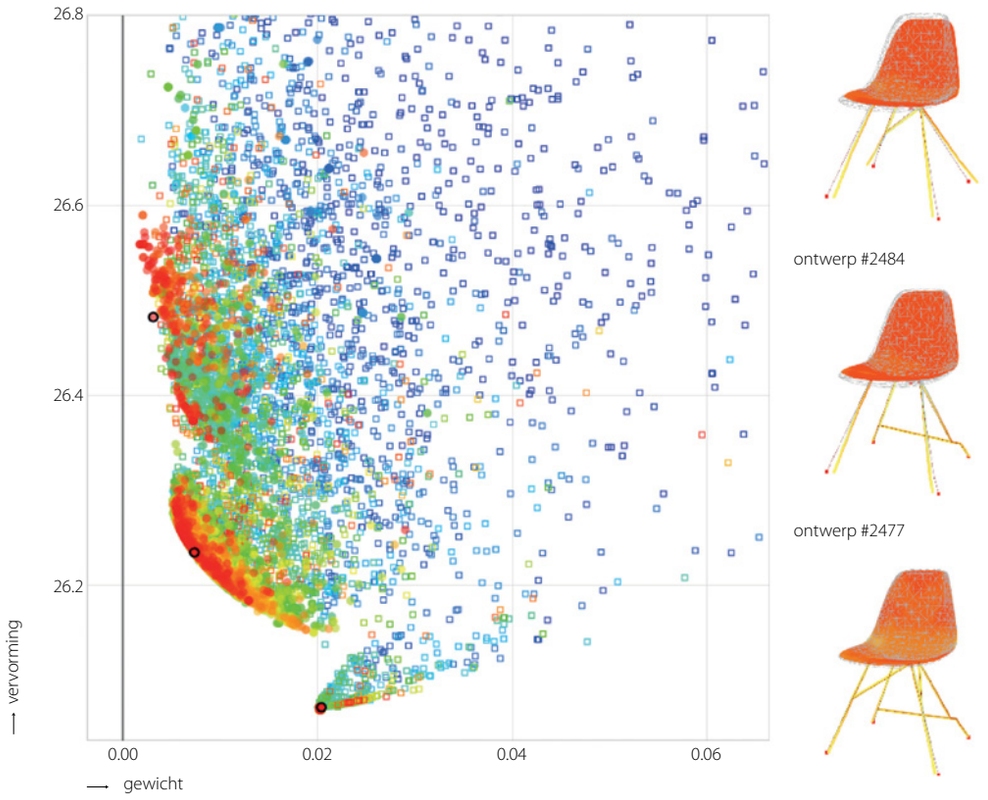
Als de ontwerpen worden geplot ten opzichte van de twee doelen, wordt de wisselwerking tussen beide duidelijk zichtbaar als een lijn die van de linkeronderhoek naar de rechterbovenhoek loopt. In dit geval is de afweging vrij eenvoudig en bijna lineair, omdat ontwerpen met meer volume ook de neiging hebben een groter oppervlak te hebben. De ontwerpen langs deze lijn kunnen allemaal als optimaal worden beschouwd, omdat er voor elk ontwerp geen ander ontwerp is dat het in beide doelen verslaat. Bijvoorbeeld, de ontwerpen # 126 en # 137 kunnen beide als optimaal worden beschouwd, maar ontwerp # 5 niet omdat er veel ontwerpen zijn die zowel meer volume als minder oppervlakte hebben.

De ontstane lijn wordt de optimale Pareto-grens genoemd, wat verwijst naar het concept van Pareto-efficiëntie in de economie. Dit optimale front is een soort grens die de ontwerpruimte in termen van zijn doelen verdeelt. Alle ontwerpen op de grens zijn optimaal omdat het onmogelijk is het ontwerp beter te maken in één doel zonder het slechter te maken in ten minste één ander doel. Terwijl je langs de grens beweegt, vind je andere ontwerpen die even optimaal zijn, maar de wisselwerking op een andere manier oplossen. Elk ontwerp dat binnen (rechts van) de grens wordt gevonden, is haalbaar maar niet optimaal, omdat je eenvoudig ontwerpen kunt vinden die beter presteren in alle doelen door simpelweg dichter bij de grens te komen. De grens markeert het einde van de ontwerpruimte.

Om de oriëntatie van de grens duidelijker te maken, kan een 'utopia-punt' in de grafiek worden gedefinieerd als de hypothetisch beste oplossing voor allebei de doelen: zo groot mogelijk volume, zo laag mogelijk oppervlak. In dit geval zou dit een object zijn met een oneindig volume en geen oppervlakte, wat zowel in dit model als in de fysieke wereld duidelijk onmogelijk is. Een manier om hier intuïtief over na te denken, is dat alle ontwerpen in de ontwerpruimte zo dicht mogelijk bij het utopia-punt willen komen, terwijl het optimale front de grens aangeeft hoe ver ze kunnen komen.



9 Discontinue optimale grens bij een discontinue ontwerpruimte



Visualisatie van optimale Pareto-grens en utopia-punt

In de praktijk is deze grens niet altijd even helder en ononderbroken als in het voorgaande. Met een discontinue ontwerpruimte zal de grens ook discontinu zijn, met verschillende delen die optimale grenzen vertonen binnen verschillende gebieden van de ontwerpruimte. Bekijk bijvoorbeeld de grafiek in figuur 9, die de ontwerpruimte van een stoel toont, door een algoritme geoptimaliseerd met twee doelen: het gewicht minimaliseren en de vervorming beperken. In dit model is er een discontinuïteit elke keer als de configuratie van de benen verandert of een nieuwe staaf wordt toegevoegd.

Om de best mogelijke ontwerpen binnen onze ontwerpruimte te vinden, is het de taak van het genetische algoritme om alle ontwerpen langs het optimale front te vinden. Hiervoor moet het algoritme verkenning combineren met optimalisatie om ervoor te zorgen dat het geen delen van de grens mist en tegelijk ontwerpen zo optimaal mogelijk maakt, zodat de grens zo dicht mogelijk bij het utopia-punt (linksonder) komt.

In het bovenstaande diagram geeft de kleur van elke cirkel het tijdstip aan waarop het door het algoritme werd verkend, waarbij blauwe ontwerpen vroeg zijn onderzocht en rode ontwerpen later in het zoekproces werden onderzocht. Hier is duidelijk te zien dat het algoritme presteert zoals het zou moeten: het vinden van meerdere gebieden van het optimale front en ze in de loop van de tijd meer en meer optimaliseren, maar toch genoeg verkennen om niet vast te zitten aan een bepaalde oplossing voor de afweging.

Willekeurig aantal doelen

De optimale grens is eenvoudig te visualiseren bij slechts twee doelen. Het concept van Generative Design geldt echter ook voor een willekeurig aantal doelen, waarbij de optimale grens van de ontwerpruimte een conceptueel n-dimensioneel hyperoppervlak is (hierbij is n het aantal dimensies).

Menselijke ontwerper

Geconfronteerd met meerdere doelen kan worden geconcludeerd dat Generative Design zelden in staat is één enkele beste ontwerpoplossing te bieden. Dit laat zien dat het doel van deze methode niet is de menselijke ontwerper te vervangen, omdat de ontwerper nog steeds het uiteindelijke ontwerp uit de pool van optimale ontwerpen moet selecteren. Het enige wat het zoekalgoritme kan doen, is een deel van de structuur van de ontwerpruimte onthullen aan de ontwerper. Hij kan vervolgens die kennis gebruiken om een ontwerp te selecteren op basis van andere doelen of voorkeuren, of de ontwerpruimte opnieuw in te kaderen en verder te verkennen.

Voorbeeld: General Motors gordelverbinding [3)]

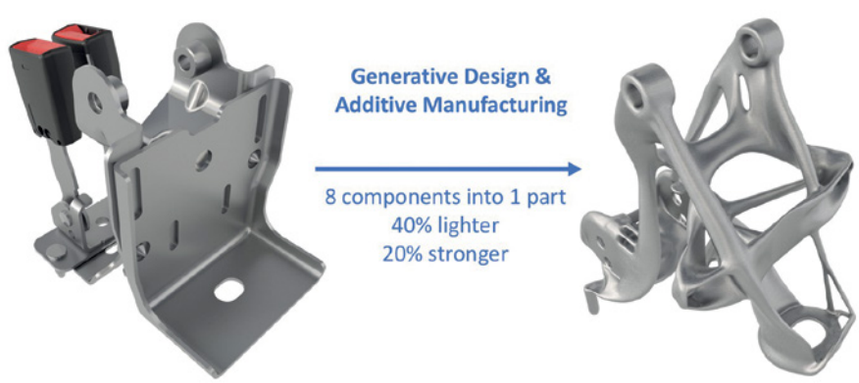
Om het gewicht van hun auto's verder te verminderen, heeft General Motors de Generative Design-software van Autodesk gebruikt. Met Generative Design kan het gewicht van onderdelen worden verminderd en kunnen combinatieonderdelen worden gemaakt die met 3D-printen kunnen worden vervaardigd.

In een proof-of-concept gebruikten ingenieurs van GM's Tech Center in Warren, Michigan, de Generative Design-software om een klein, maar belangrijk onderdeel te herontwerpen: de beugel waar veiligheidsgordels aan worden vastgemaakt. De software produceerde meer dan 150 geldige ontwerpopties op basis van parameters die de ingenieurs instelden, zoals vereiste verbindingspunten, sterkte en massa (fig. 10). Ze richtten zich op een nieuw ontwerp, waarvan de organische structuur niet door een ontwerper had kunnen worden bedacht. Het resultaat is 40% lichter en 20% sterker dan het originele onderdeel.

Dit voorbeeld toont een ander groot voordeel van Generative Design: consolidatie van onderdelen. Het nieuwe onderdeel combineert acht verschillende componenten in één 3D-geprinte beugel. Dat betekent een besparing van kosten op zowel economisch als op milieuvlak. Naast het vervoer vervallen ook de kosten van het lassen of in elkaar schroeven van de onderdelen. Een 3D-printer is genoeg.

Hoewel deze beugel nog steeds een prototype is, is het potentieel van Generative Design enorm. Dezelfde techniek kan voor veel meer delen van auto's worden gebruikt, maar ook voor de (staal) constructies van gebouwen en bruggen. We zijn pas net begonnen met het verkennen van de mogelijkheden.

10



11

10 Enkele ontwerpopties voor de stoelbeugel
bron: General Motors

11 Traditioneel versus Generative Designed onderdeel
bron: General Motors

[3)] *Gebaseerd op een artikel van Bill Danon [3].*

12

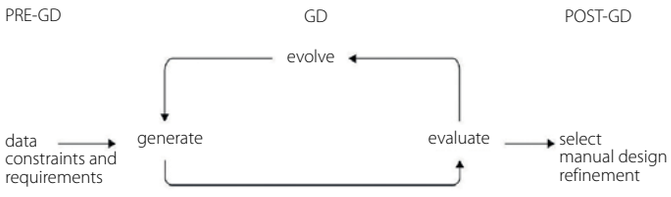
Voorbeeld: Autodesk University (AU) 2017 Expositiehal indeling [4)]

Voor Autodesk University Las Vegas (AU) moest een nieuwe indeling voor de expositiehal worden gevonden (foto 12). De uitdaging bestond uit het verdelen van alle verschillende programma's binnen de expositieruimte, terwijl de beperkingen en voorwaarden werden gerespecteerd en belichtingsniveaus en aandacht ('buzz') moesten gemaximaliseerd. In het verleden was de indeling van de AU-expositiehal gebaseerd op vuistregels en menselijke ontwerpintuïtie. Bijvoorbeeld het gebruik van symmetrische lay-outs met een focus op het geometrische centrum van de hal en het samenbrengen van clusteringprogramma's of het plaatsen ervan in de nabijheid van toegangspunten. Dit garandeert, volgens eerdere ervaringen, het juiste niveau van aandacht.
Door de toepassing van Generative Design werd aan deze traditionele lay-outs en vuistregelbesluitvorming van eerdere jaren voorbijgegaan om nieuwe lay-outoplossingen te ontdekken, die zowel vernieuwend zijn als goed presteren op de gegeven ontwerpdoelen.

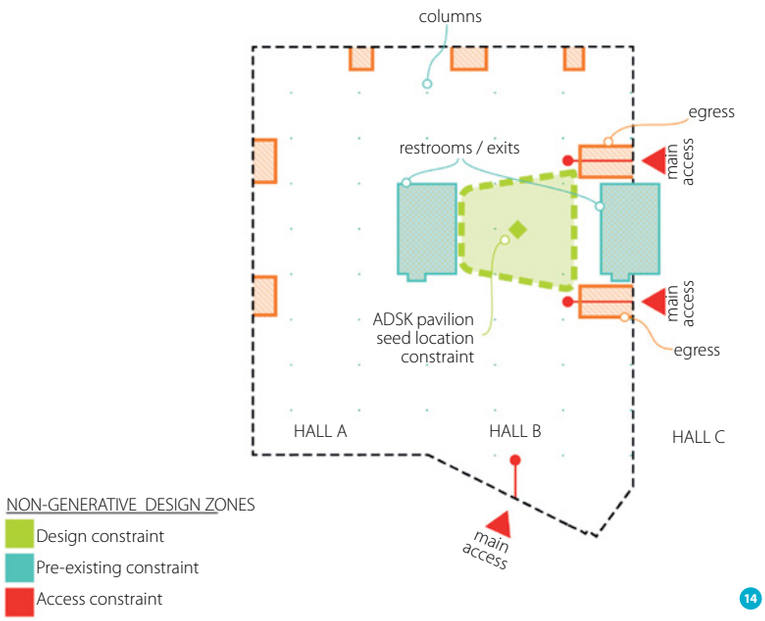
Generative Design (GD) wordt voorafgegaan door een fase die 'pre-generatief ontwerp' (pre-GD) wordt genoemd en wordt

4) Gebaseerd op een artikel van Lorenzo Villaggi en Danil Nagy [2].

12 Autodesk University Las Vegas
13 Generative Design (GD) wordt voorafgegaan door pre-generatief ontwerp en gevolgd door post-generatief ontwerp
14 Randvoorwaarden van de AU-expositiehal



13



14

gevolgd door een fase die 'post-generatief ontwerp' (post-GD) wordt genoemd.

Pre-GD

De pre-GD-fase omvat het nauw samenwerken van de belanghebbenden om unieke en kritieke gegevens (waarden) te verzamelen over het project als input voor het generatieve model en de evaluatieve component.

Gegevens verzamelen: vereisten en beperkingen

De eerste stap is het verzamelen van gegevens. Informatie over relatieve nabijheid en locatievoorkeuren wordt verzameld.
De verzamelde randvoorwaarden zijn de volgende (fig. 14):
– ontwerpvoorwaarden – plaatsing van een deel van de Autodesk-paviljoens naast de hoofdtoegangspunten van de keynote-hal;
– al bestaande voorwaarden: de begrenzing van de expositiehal, locatie van de kolommen, uitgangsgebieden en centrale toiletten;
– toegangsvoorwaarden: belangrijke vaste locaties voor toegang en vertrek van de hoofdruimte.

Formulering doelstellingen

De tweede stap is het formuleren van doelen. Samen met de betrokkenen werden twee ontwerpdoelen bepaald:
– aandacht ('buzz') – als waarde voor de hoeveelheid en verdeling van activiteit, weergegeven als een web van rode lijnen met variërende dikte die de hoeveelheid verkeer aangeeft;
– zichtbaarheid – als waarde voor de nabijheid van de stands tot hoge activiteitszones, geïllustreerd door kleurcodering van de cabines met een gradiënt die van wit naar rood gaat, waarbij wit een lage zichtbaarheid en rood een hoge zichtbaarheid weergeeft.

Generative Design

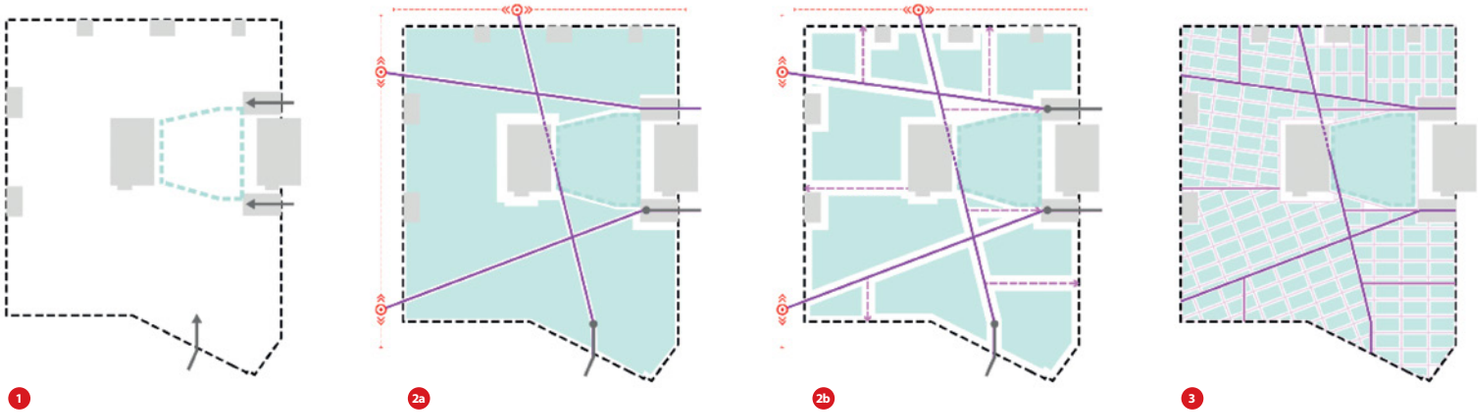
GD bestaat uit drie hoofdcomponenten:
1. een generatief model dat een brede ontwerpruimte van mogelijke oplossingen kan beschrijven;
2. een evaluatieve component die de gespecificeerde ontwerpdoelen omvat;
3. een evolutionair metaheuristisch zoekalgoritme, in dit geval een Genetisch Algoritme, dat door de ontwerpruimte kan navigeren en steeds betere ontwerpoplossingen genereert.

Het proces van Generative Design bestaat uit de stappen genereren (1), evalueren (2) en evolueren (3).

1. Genereer

Het geometriesysteem voor de expositiehal is geïnspireerd op de stedelijke morfologie en de wijze waarop steden groeien. Deze strategie kan worden toegepast op zeer uiteenlopende situaties: van onregelmatige indelingen die lijken op historische buurten in Europese steden zoals Rome of Parijs, tot meer regelmatige indelingen zoals de stad New York.

De ontwerpruimte en ontwerpmethodologie voor het genetisch algoritme is als volgt (fig. 15):
1. Definieer geometrische beperkingen en niet-generatieve ontwerpzones.
2a. Genereer drie of meer belangrijke lanen die de tentoonstellingshal onderverdelen in macroregio's.
2b. Verdeel elk macroregio in twee subregio's, door een reeks secundaire lanen.
3. Verdeel elk deelgebied onder in cellen via een raster met een verschillende oriëntatie.
4. Plaats belangrijke 'programmaspots' langs de lanen.
5. Laat 'spots' groeien door aangrenzende cellen te verenigen.
6. Vul de resterende cellen met standaardstands.



2. Evalueer

Via de geautomatiseerde evaluatieve componenten wordt elk ontwerp gescoord langs de twee vastgestelde doelen: aandacht ('buzz') en zichtbaarheid (fig. 16). Dergelijke numerieke waarden worden door het zoekalgoritme gebruikt om goed presterende ontwerpen te ontwikkelen.

3. Evolueer

Voor dit project zijn meer dan 30.000 ontwerpen gegenereerd: 100 generaties vermenigvuldigd met 320 ontwerpoplossingen. Zoals te zien is in figuur 16, heeft de evolutionaire component geleerd de inputs op een zodanige manier te besturen dat verschillende families van goed presterende ontwerpen worden gevonden.

Post-GD

In de post-GD-fase wordt de menselijke component kritiek. De menselijke ontwerper kan de grens verkennen van goed presterende ontwerpen (Pareto-voorkant), compromissen bestuderen en ontwerpopties kwalitatief beoordelen.

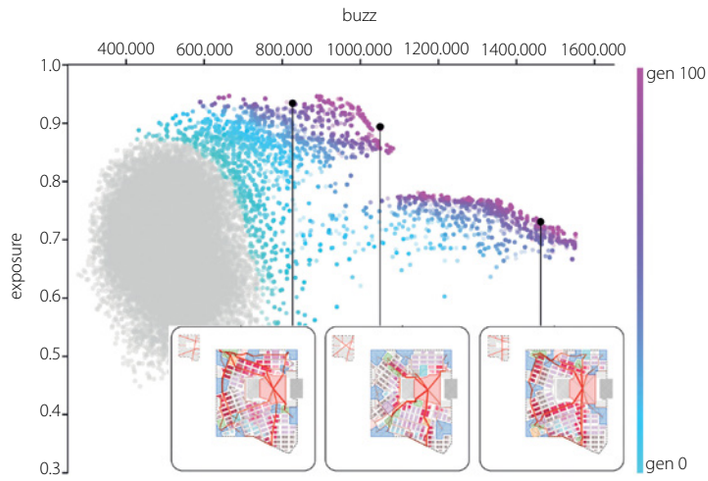
Selecteren

Door een subset van goed presterende ontwerpen direct te inspecteren, kan de ontwerper samen met de belangrijkste belanghebbenden een klein aantal kandidaat-ontwerpen selecteren om handmatig te verfijnen (fig. 17).

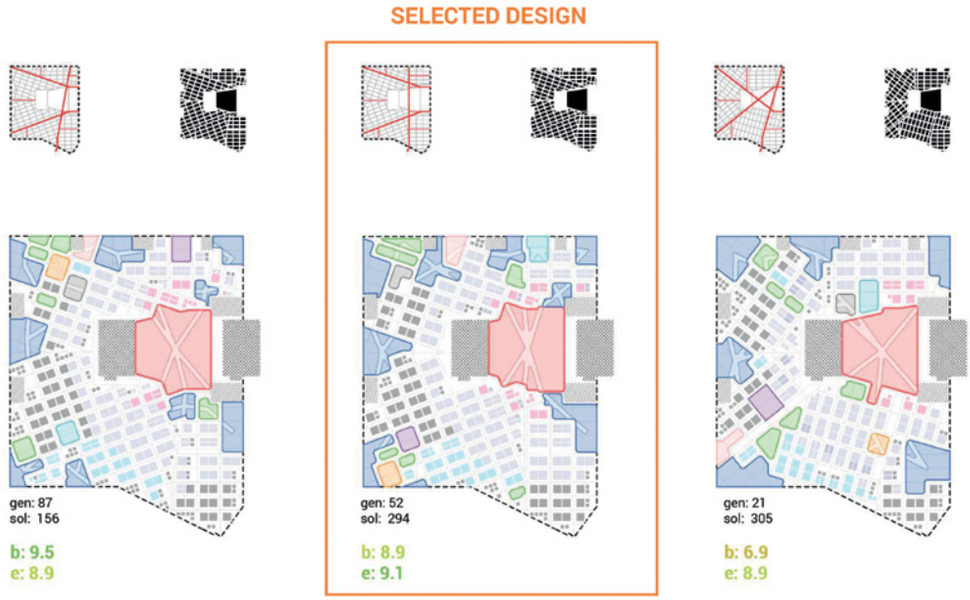
Verfijnen

Nadat de selectie de zoekopdracht heeft beperkt tot een handvol ontwerpen, vindt handmatige verfijning plaats om het ontwerp verder te ontwikkelen en ervoor te zorgen dat aan de beperkingen en vereisten wordt voldaan (fig. 18). In dit geval, als een post-GD-ontwerpkenmerk, zijn de paden met veel verkeer gebruikt om primaire circulatieroutes te genereren die ook kunnen helpen met vindbaarheid. Op een vergelijkbare manier worden de voorspelde drukke paden gebruikt om de verschillende hoofdposten van Autodesk te verdelen, die op hun beurt bezoekers van de hoofdtoegangspunten naar de industriestands kunnen leiden.

16



17



18

15 Ontwerpmethodologie
16 Ontwerpdoelen
17 Beoordelen en selecteren van ontwerpen in de ontwerpruimte
18 Drie handmatig verfijnde goed presterende ontwerpen

Conclusie

Generative Design is een nieuwe ontwerpmethode die het mogelijk maakt onverwachte nieuwe ontwerpen te ontdekken en trade-offs te onderzoeken tussen beperkingen en doelen bij goed presterende ontwerpen. Een dergelijke methode introduceert nieuwe manieren om architectuur te bedenken, te maken en te produceren, samenwerking tussen ontwerpers en klanten te stroomlijnen en tegelijkertijd de creatieve kracht van kunstmatige intelligentie voor ontwerpers en ingenieurs te ontsluiten. ☒

Bronnen

1 Nagy, D. (2017). *Evolving design*. Generative design, https://medium.com/generative-design/evolving-design-b0941a17b759.

2 Viallagi, L., Nagy, D. *Generative Design for Architectural Space Planning: The Case of the AU 2017 Exhibit Hall Layout.* Autodesk University, https://medium.com/autodesk-university/generative-design-for-architectural-space-planning-9f82cf5dcdc0.

3 Danon, B. (2018) *How GM and Autodesk are using generative design for vehicles of the future.* Autodesk, https://adsknews.autodesk.com/news/gm-autodesk-using-generative-design-vehicles-future.

4 Holland, J. (1975) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* Michigan: University of Michigan Press.

5 Shiffman, D. (2012) *The nature of Code: Simulating Natural Systems with Processing*.