

White Paper

Optimaal ontwerp
van constructies

AutoDesign en parametrische optimalisatie

Doorbraaktechnologie in Scia Engineer 2008

Lubomir Sabatka, Martin Novak, Emiel Peltenburg, J.P. Rammant

Inhoud

- Inleiding
- Optimalisatie van systemen
- Vijf voorwaarden voor het optimaal ontwerp van constructies
- AutoDesign en parametrische optimalisatie in Scia Engineer
- Praktijkvoorbeelden
- Doorgedreven onderzoek
- Voordelen voor bouwkundige ingenieurs
- Referenties

Inleiding

De meeste CAE softwareprogramma's spiegelen u voor dat zij u helpen bij het creëren van een optimaal ontwerp van uw constructies. Maken zij in werkelijkheid deze beloften ook waar?

Als men gewoonlijk over de optimalisatie van constructies spreekt, bedoelt men in feite het optimaliseren van enkele geselecteerde onderdelen van de constructie. Een ontwerpingenieur is bijvoorbeeld op zoek naar de juiste minimale afmetingen voor een doorsnede die de vereisten van de ontwerpcode moet vervullen, hij of zij tracht het minimumaantal benodigde bouten voor een specifieke staalverbinding te vinden, zoekt de minimaal vereiste wapeningszone in een betonnen balk enz. Hoewel alle structuuronderdelen zo optimaal mogelijk ontworpen worden, betekent dit niet automatisch dat de volledige constructie optimaal is, bijvoorbeeld als men haar vanuit het standpunt van de materiaalkosten, de benodigde bouwtijd, de loonkosten, enz. beschouwt.

Het optimale ontwerp van een constructie vindt men na het zorgvuldig testen en vergelijken van vele varianten. Iedereen is het daarmee eens, maar hoe vaak gebeurt dit ook daadwerkelijk in de bouwnijverheid? De ontwerper werkt altijd onder grote druk van de klant en meestal is er niet veel tijd om varianten uit te werken.

Een typisch voorbeeld is een balk in gewapend beton. De afmetingen van de doorsnede worden gekozen, de interne krachten worden berekend en vervolgens wordt de wapening – optimaal, uiteraard – ontworpen. Maar wie denkt er aan om een beetje met de hoogte en de breedte van de balk te spelen en zo een optimale prijs voor de volledige balk te zoeken op basis van de prijs van het beton en het staal?

De bestaande softwaremiddelen bieden nochtans iedereen de mogelijkheid om dit te doen; het is alleen maar een kwestie van een aantal varianten uit te proberen en te vergelijken om de meest geschikte te vinden. Dit zou realistisch zijn voor kleinere projecten, maar zeker niet voor uitgebreide en complexe. Een ervaren ontwerper zal natuurlijk dichter bij een optimale oplossing komen dan een beginner.

Optimalisatie van systemen

Het onderzoek naar optimalisatie wordt voornamelijk gevoerd in de automotive sector en de ruimtevaartindustrie, met de nadruk op het berekenen van de vloeistofdynamica en de structurele optimalisatie, hoofdzakelijk van de vorm.

Er bestaan talloze mathematische methoden die perfect aan de behoeften van de bouwindustrie beantwoorden. Het systeem wordt doorgaans beschreven aan de hand van een aantal parameters. Vervolgens moet men het doel van de optimalisatie bepalen – wat is de bedoeling, wat moet er geminimaliseerd of gemaximaliseerd worden. En dit doel moet met eender welke set parameters berekend kunnen worden. Op die manier wordt de “doelfunctie” of de “prijsfunctie” bepaald.

Na het definiëren van een parameterset en de doelfunctie kan men de gebruikelijke mathematische optimalisatiemethodes gebruiken, ongeacht of men het gewicht van een spaceshuttle of het verkeer op straat optimaliseert.

Vijf voorwaarden voor het optimaal ontwerp van constructies

Als er krachtige softwaremiddelen en duidelijke mathematische methoden beschikbaar zijn voor het ontwerpen van structurele onderdelen en de optimalisatie van systemen, waarom is de optimalisatie van constructies dan geen ingeburgerde gewoonte?

De huidige CAE softwaresystemen zijn onvoldoende uitgerust voor de structurele optimalisatie. Zij zouden toch een aantal vereiste functies moeten integreren. Een overzicht:

Voorwaarde 1

Functies voor het optimale ontwerp van specifieke structuuronderdelen zoals staalbalken, betonbalken, staalverbindingen, funderingsblokken, enz. Gewoonlijk is men op zoek naar de minimumafmetingen, -maten of aantallen. Het onderdeel moet bovendien voldoen aan de criteria van de bijbehorende code.

Voorwaarde 2

Het programma moet de mogelijkheid bieden de constructie te parametriseren. De ontwerper moet de vaste en de variabele onderdelen in de constructie bepalen – overspanningen, hoogtes, afmetingen van doorsneden, dikte van platen en muren, belastingen... Ieder element dat kan variëren moet met behulp van een onafhankelijke parameter beschreven kunnen worden. Andere afmetingen kunnen afhangen van parameters en zo een intelligent parametrisch constructiemodel mogelijk maken.

Voorwaarde 3

De doelfunctie moet gedefinieerd kunnen worden. Dit kan het gewicht van het benodigde staal, het volume beton, het gewicht van de wapening, maar ook de maximale verplaatsing of eender wat zijn. De perfecte situatie is als het systeem een algemene waarde zoals bv. de prijs kan berekenen.

Voorwaarde 4

Het softwaresysteem moet de doelfunctie voor een specifieke set parameters kunnen evalueren. Dit betekent dat er een functie aanwezig moet zijn, die de parametersets kan lezen en een doelwaarde kan genereren.

Voorwaarde 5

Er moet een optimalisatie solver beschikbaar zijn. Dit is een tool dat verschillende parametersets genereert, de doelfunctie berekent en uiteindelijk de optimale set parameters voorstelt.

Als een CAE systeem over deze functies beschikt, ligt de weg naar de optimalisatie open.

AutoDesign en parametrische optimalisatie in Scia Engineer

Scia Engineer 2008 integreert alle vereiste functies, die hierboven beschreven werden. De functionaliteit die nodig is voor de vijf eerder genoemde voorwaarden wordt in vijf stappen gerealiseerd. We gebruiken het woord "AutoDesign" in de eerste stap van het optimale ontwerp van de constructie-elementen te beschrijven. Het woord "optimalisatie" gebruiken we in stap vijf, het is de volledige oplossing voor de ganse constructie.

Stap 1 - AutoDesign

Staal- en betonbalken kunnen optimaal ontworpen worden. De optimalisatie van stalen doorsneden is al jarenlang ingeburgerd; het optimale ontwerp van wapening werd pas recentelijk beschikbaar gemaakt. Ook diverse staalverbindingen kunnen geoptimaliseerd worden en het ontwerp van de minimaal vereiste wapening in platen en muren is beschikbaar. Met Scia Engineer kunnen alle benodigde elementoptimalisaties gedefinieerd worden; het programma onthoudt ze en herhaalt ze na iedere wijziging van de invoergegevens van een constructie.

Stap 2 - Parameters

Het parametreren van de constructie is een basismogelijkheid voor het modelleren in Scia Engineer. Nagenoeg iedere entiteit, ieder element van het ontwerpproject kan met een gebruikergedefinieerde parameter gedefinieerd worden. Parameters worden aan variabelen toegewezen, te beginnen met de afmetingen van de structurele elementen en doorsneden, belastingen en massa's, via de benodigde tijd voor het storten van het beton tot en met bijvoorbeeld de diameter of de dekking van de staalwapening.

Stap 3 – Invoer/uitvoer XML interface

Scia Engineer is eveneens uitgerust met een algemene XML tekstinterface, waarmee de structurele gegevens van "buitenaf" gewijzigd kunnen worden en iedere benodigde waarde voor een project ingelezen kan worden. De gebruiker definieert in het XML document de te integreren ontwerpwaarden. Alle Scia uitvoerdocumenten zijn "levende" documenten, die geüpdatet kunnen worden na het wijzigen en herberekenen van het project. Het XML document vervult de rol van de doelfuncties zonder enige beperking. Het praktijkvoorbeeld hieronder toont hoe het XML bestand gedefinieerd wordt.

Stap 4 – ESA XML solver

Scia Engineer werkt ook in "hidden" modus. Het project kan gewijzigd worden, het berekeningsproces kan gestart worden, alle gedefinieerde documenten kunnen hernieuwd en geüpdatet worden en alle waarden in het document kunnen ingelezen worden en dat allemaal vanuit een ander programma dan Scia Engineer. Om dit te realiseren, is er een extra toepassing, ESA_XML.exe, die gemakkelijk op een externe softwaretoepassing aangesloten kan worden.

Stap 5 – Parametrische optimalisatie

De eenvoudigste optimalisatie solver is diegene die alle mogelijke sets invoerparameters genereert en een doelfunctie voor iedere set berekent. Op die manier wordt de minimum (of maximum) doelwaarde gevonden met de optimale set invoerparameters. Deze werkwijze is eenvoudig en

betrouwbaar. Als men alle mogelijke varianten berekent, vindt men uiteindelijk de meest geschikte. Het enige probleem bij systemen met vele parameters is een drastische stijging van het aantal varianten.

Dit soort “batch optimiser” is nu beschikbaar in Scia Engineer. De gebruiker moet alleen maar de limieten en de variatiestap voor de parameters definiëren. Alle varianten worden berekend en de diagrammen met de resultaten worden in Excel gegenereerd.

De laatste functie voor het optimaliseren is de echte optimalisatie solver. Deze solver maakt gebruik van dezelfde invoer als de “batch optimiser” – met de limieten en stappen voor de parameters. Hier worden echter niet alle varianten berekend, er worden slechts enkele “voorgesteld”; de optimale set parameters wordt afgeleid uit de resultaten van de doelfunctie. De magische van de kwaliteit van deze solver is zijn uitstekende “voorstellingsvermogen”. Hoe beter de solver, hoe minder variantenberekeningen er vereist zijn.

Scia werkt voor dit onderwerp samen met de Universiteit voor Burgerlijke Bouwkunde van Praag. In de huidige stand van zaken wordt de optimalisatie solver van de universiteit aangesloten op Scia Engineer. De solver gebruikt een stochastisch algoritme van “Gesimuleerde Annealing” op basis van algemene genetische algoritmes. Hierdoor wordt ook een multi-parameter optimalisatie mogelijk, waardoor meer resultaatwaarden gecontroleerd kunnen worden. Deze methode garandeert dat er meer “optimale” oplossingen gevonden worden, want zij zoekt lokale extremen die zinvol zijn vanuit het ontwerpstandpunt.

Praktijkvoorbeelden van optimalisatie

Er zijn 4 verschillende soorten structurele optimalisatie:

- **Topologie**optimalisatie, dit betekent een constructie vinden zonder haar uiteindelijke vorm te kennen; dit houdt in dat er tijdens de berekening van de variantenoplossingen elementen of FEM netonderdelen verwijderd/toegevoegd worden
- **Vorm**optimalisatie: de topologie van de constructie is vanaf het begin gekend, maar er zijn bepaalde onderdelen, waar bijvoorbeeld hoge spanningen problemen kunnen veroorzaken. De vormparameters worden dus geoptimaliseerd om de spanningen te minimaliseren.
- **Afmetingen**optimalisatie: een constructie wordt gedefinieerd door een set afmetingen, maten of doorsneden; deze worden gecombineerd om de gewenste optimale criteria te behalen.
- **Topografie**optimalisatie, dit betekent een geschikte vorm vinden voor een constructie (bijvoorbeeld tent, membraan, brug).

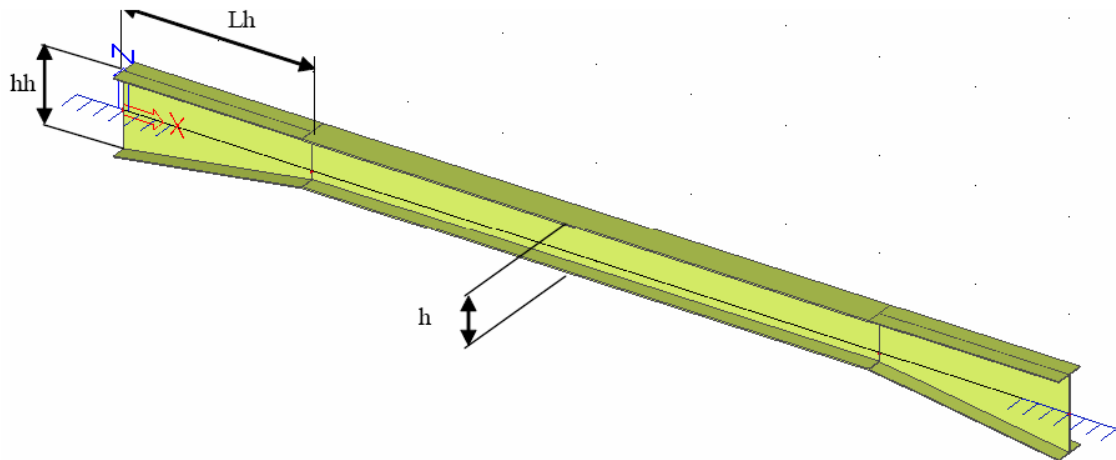
Vele voorbeelden van structurele optimalisatie zijn manifest aanwezig in de dagdagelijkse ontwerppraktijk, we noemen er enkele op:

- De optimale verhouding tussen de stijfheden van balken en kolommen zoeken
- De optimale dikte voor betonplaten vinden
- De optimale afmetingen voor betonbalken bepalen
- De optimale vorm voor een nagespannen voorspanelement vinden
- Optimalisatie van de positie van funderingspalen
- Gevoeligheidsanalyse van de parameters van de ondergrond
- De meest prijsgunstige staalverbinding ontwerpen
- Het gewicht van een staalconstructie voor een vooraf bepaald type frame minimaliseren
- De optimale definitie voor brugoverspanningen zoeken
- De maximale draaglast voor een kraan in diverse geometrische posities vinden
- ...

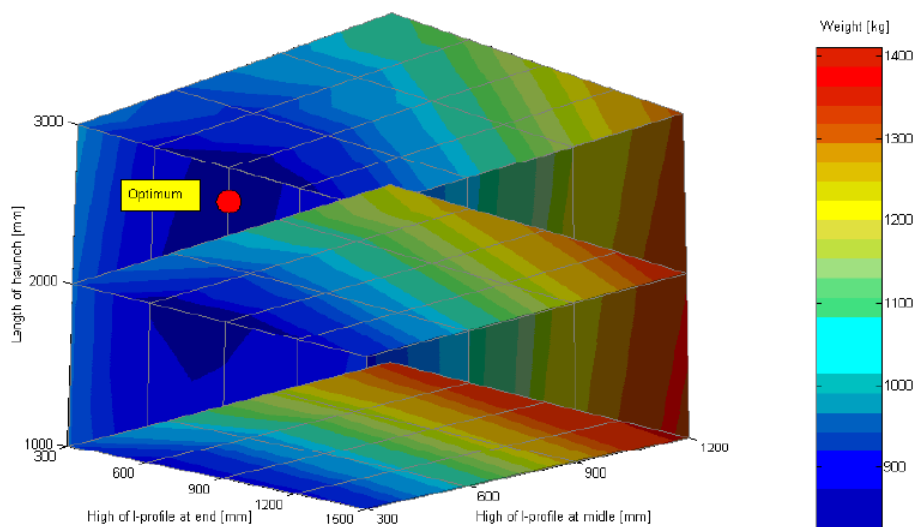
Om de discrete parametrische batchverwerking in Scia Engineer te illustreren, zetten we enkele kleine voorbeelden uiteen.

Voorbeeld 1: Optimalisatie van de lengte en diepte van een boog van een versmalde balk

We nemen een balk met boog; de balk bestaat uit gelaste I-profielen. De diepte van de balk in het midden en aan het einde plus de lengte van de boog worden geparametreerd. De flens- en lijfdikte wordt automatisch ontworpen in elke stap van het optimalisatieproces.



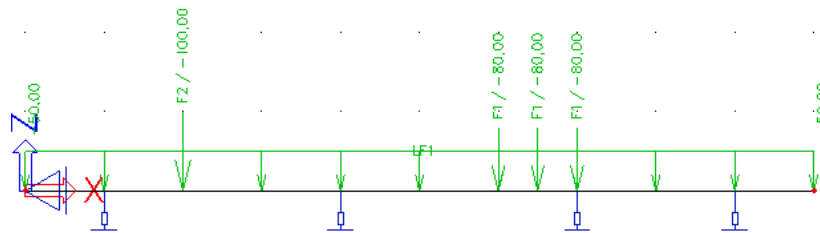
Hoe U zult merken dat alle varianten voldoen aan de vereisten van de norm, toch varieert het gewicht binnen een ruime schaal.



Er werd een optimale vorm van de versmalde balk bepaald.

Voorbeeld 2: Optimalisatie van de positie van palen.

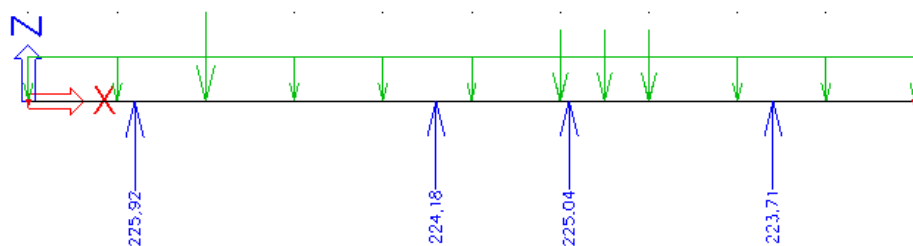
De gebruiker moet een door palen ondersteunde funderingsbalk ontwerpen. In het ideale geval is de reactiekracht in alle palen hetzelfde.



De gebruiker kan de limieten voor de verplaatsing van de steunen bepalen, iedere variante wordt berekend en de reacties worden geëvalueerd. De doelfunctie kan een eenvoudige formule zijn, die de kwadratensom van afwijkingen van een gemiddelde reactiewaarde berekent.

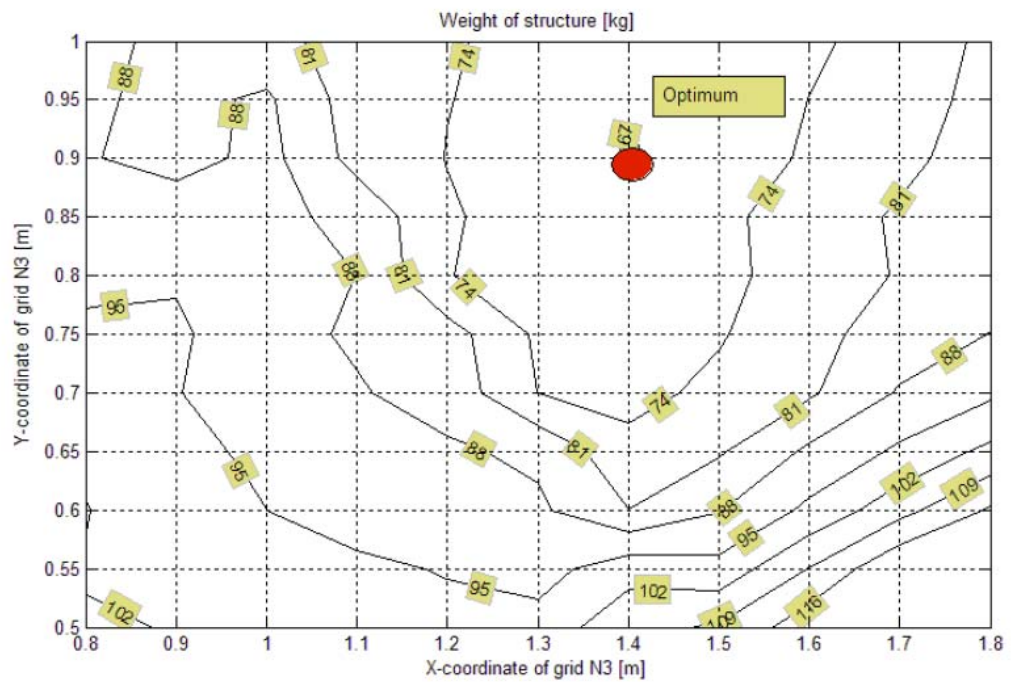
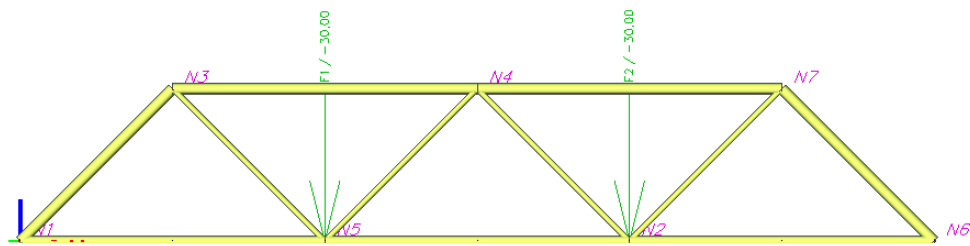
L1 [m]	L2 [m]	L3 [m]	L4 [m]	R1 [kN]	R2 [kN]	R3 [kN]	R4 [kN]	R [kN]
1,1	4,6	6,1	8,2	217,397	221,908	225,697	233,858	12,083
1,1	4,6	6,1	8,3	219,58	223,745	226,094	229,441	7,18
1,1	4,6	6,1	8,4	221,58	225,586	226,678	225,015	3,812
1,1	4,6	6,2	8,2	219,897	223,372	225,472	230,119	7,402
1,1	4,6	6,2	8,3	222,036	225,238	225,811	225,776	3,127
1,1	4,6	6,2	8,4	223,997	227,104	226,342	221,418	4,443
1,1	4,7	6,1	8,2	220,634	222,376	224,541	231,309	8,102
1,1	4,7	6,1	8,3	222,825	224,165	224,958	226,913	2,961
1,1	4,7	6,1	8,4	224,833	225,964	225,556	222,508	2,675
1,1	4,7	6,2	8,2	223,167	223,803	224,279	227,611	3,436
1,1	4,7	6,2	8,3	225,314	225,623	224,637	223,286	1,797
1,1	4,7	6,2	8,4	227,283	227,448	225,183	218,946	6,896
1,2	4,6	6,1	8,2	221,671	220,56	224,111	232,517	9,368
1,2	4,6	6,1	8,3	223,888	222,369	224,484	228,119	4,222
1,2	4,6	6,1	8,4	225,921	224,182	225,043	223,714	1,686
1,2	4,6	6,2	8,2	224,201	221,999	223,874	228,786	4,992
1,2	4,6	6,2	8,3	226,373	223,836	224,188	224,463	1,966
1,2	4,6	6,2	8,4	228,366	225,673	224,695	220,126	5,942
1,2	4,7	6,1	8,2	224,954	220,993	222,953	229,96	6,673
1,2	4,7	6,1	8,3	227,179	222,753	223,345	225,584	3,543
1,2	4,7	6,1	8,4	229,219	224,523	223,918	221,2	5,772
1,2	4,7	6,2	8,2	227,516	222,395	222,678	226,271	4,45
1,2	4,7	6,2	8,3	229,697	224,185	223,011	221,967	5,963
1,2	4,7	6,2	8,4	231,697	225,981	223,532	217,649	10,084

Als we deze functie minimaliseren worden alle palen met dezelfde kracht belast.

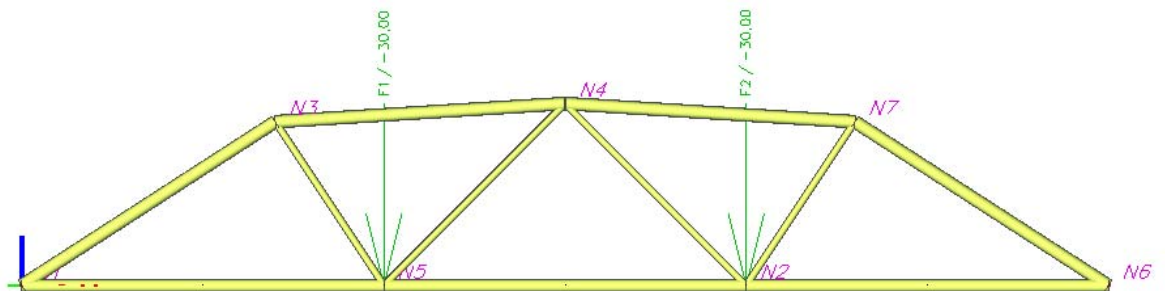


Voorbeeld 3: Vormoptimalisatie van een eenvoudig raamwerk

De vorm van de eenvoudige vakwerklijger wordt onderzocht. Parameters zijn de coördinaten van de knopen N3, N4 en N7 die de symmetrie nauwkeurig respecteren. Het doelcriterium wordt gekozen, bv. minimum gewicht.



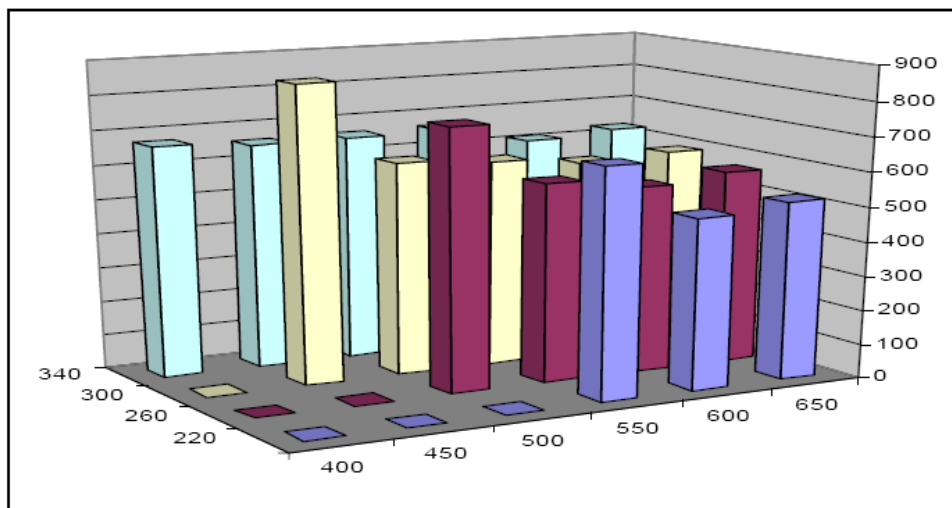
De optimale vorm van de constructie is als volgt:



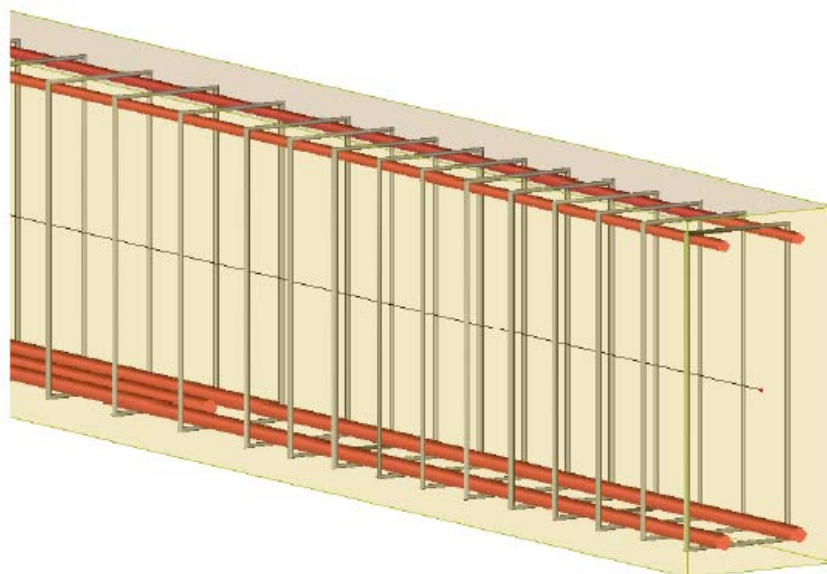
Voorbeeld 4: Optimalisatie van de afmetingen van de doorsnede van een gewapende betonbalk

De 8-meter lange betonbalk die met 50kN/m geladen is, wordt onderzocht. De diepte van de doorsnede varieert van 400 tot 650mm, de breedte van 220 tot 400mm. De doelfunctie is de prijs van een balk – deze wordt berekend via de staal- en betonprijs. De optimale vorm hangt natuurlijk af van de prijsbepalingen per land. In dit geval gebruiken we 150€ per m³ beton en 2€ per kilo wapening.

Depth	Width			
	220	260	300	340
400	0	0	0	671
450	0	0	860	658
500	0	756	620	660
550	663	578	602	662
600	498	545	580	613
650	520	569	592	629



De optimale vorm van de doorsnede is 600x220mm.



Lopend onderzoek

De Universiteit voor Burgerlijke Bouwkunde van Praag heeft een uitgebreid onderzoeksprogramma over optimalisatie. Het onderzoek spitst zich toe op algoritmes die geschikt zijn voor praktische bouwkundige problemen met typische discrete invoerfuncties (bijvoorbeeld een set beschikbare materialen, gewalste profielen of stalen buizen), samen met een gecompliceerde afhankelijkheid van de uiteindelijke doelfunctie van deze invoervariabelen. Omdat een constructie bovendien aan de meest uiteenlopende criteria moet voldoen, moet het ook mogelijk zijn zowel problemen als beperkingen met meerdere criteria te behandelen, die zich meestal moeilijk in sluitende mathematische vorm laten beschrijven. Diverse optimalisatiealgoritmes werden al ontwikkeld, maar de optimalisatie van real-life structuren vraagt nog altijd een zeer lange berekeningstijd. Methoden die op dit vlak een doorbraak forceren zijn gebaseerd op numerieke benaderingen, die het aantal berekeningen minimaliseren – de zogeheten “response surface” methoden. De artificiële intelligentie maakt gewoonlijk gebruik van methoden zoals neurale netwerken, zoals het zogeheten “radiale basisfunctienetwerk”.

Een van de voordelen van de hier voorgestelde optimalisatiemethodes is dat zij niet de globale extreme van een doelfunctie zoeken, maar lokale extremen en dat er dus meer lokale extremen gevonden worden. Ieder van deze extremen heeft een bepaalde betekenis en de ontwerper moet deze varianten beoordelen vanuit praktisch standpunt, vanuit het standpunt van de bouwkundige aspecten, enz.

Een andere aanpak voor het beperken van de lange berekeningsduur is de methode van de parallelle of gespreide berekening. De meeste gebruikers van Scia software beschikken over een computernetwerk met meerdere Pc's die minder dan 24 uur per dag gebruikt worden. Vaak worden ze overdag en in ieder geval 's nacht uitgeschakeld. Vooral 's nachts voeren ze meestal geen echte berekeningsopdrachten uit. Algoritmes die deze vrije capaciteit kunnen benutten zijn dus een nuttig hulpmiddel. Het scenario gaat als volgt: de ontwerper – die zich in een bepaald stadium van ontwerp van een constructie bevindt – definieert de bereiken, beperkingen, vereisten enz. Het optimalisatiealgoritme wordt 's nachts gestart en als de ontwerper 's anderendaags 's morgens terug op kantoor komt, kan hij een of meer varianten voor de ontworpen constructie kiezen, die als optimale oplossingen berekend werden.

Voordelen voor bouwkundige ingenieurs

Het optimaal ontwerpen van structuren zal het ontwerpproces ingrijpend veranderen; een “toekomstdroom” wordt werkelijkheid. Aangezien de mathematische methoden zich constant verder ontwikkelen en de snelheid van de computers almaar toeneemt, verhoogt de optimalisatie de kwaliteit van het praktische ontwerpproces in hoge mate.

Referenties

- Cantú-Paz, E. (2001). *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers.
- Matouš, K., Lepš, M., Zeman, J., Šejnoha, M. (2000). Applying genetic algorithms to selected topics commonly encountered in engineering practice. *Computer Methods in Applied Mechanics and Engineering*, 190(13-14): 1629-1650.
- Lepš, M., Šejnoha, M. (2003). New approach to optimization of reinforced concrete beams. *Computers & Structures*, 81(18-19): 1957-1966.
- Hrstka, O., Kučerová, A., Lepš, M., Zeman, J. (2003). A competitive comparison of different types of evolutionary algorithms. *Computers & Structures*, 81(18-19): 1979-1990.
- Hrstka, O., Kučerová, A. (2004). Improvements of real coded genetic algorithms based on differential operators preventing the premature convergence. *Advances in Engineering Software*, 35(3-4):237-246.
- Zitzler, E., Laumanns, M., Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In Giannakoglou, K., Tsahalis, D., Periaux, J., Papailou, P., and Fogarty, T., editors, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, Athens, Greece.
- Kukkonen, S., Lampinen, J. (2004). Comparison of generalized differential evolution to other multi-objective evolutionary algorithms. In Neittaanmäki, P., Rossi, T., Korotov, S., Onate, E., Périaux, P., and Knórzer, D., editors (2004). *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004)*, Jyväskylä.
- Lepš, M.: *Single and multi-objective Optimization in Civil Engineering with Applications*, PhD Thesis, Technical University Prague, 2004