

Genetiska algoritmer och deras tillämpningar inom konstruktion och design

**Av Axel Nordin, vid institutionen för maskinkonstruktion, Lunds
Tekniska Högskola, våren 2009**

Handledare: Damien Motte, institutionen för maskinkonstruktion

Examinator: Robert Bjärnemo, institutionen för maskinkonstruktion

Förord

Detta examensarbete har utförts på institutionen för maskinkonstruktion vid Lunds Tekniska Högskola under våren 2009. Målet har varit att visa på möjligheterna med evolutionära optimeringsmetoder inom konstruktion och design och lägga grunden till framtida studier inom området.

Stort tack för många värdefulla kommentarer och intressanta diskussioner till Damien Motte på institutionen för maskinkonstruktion som varit handledare för examensarbetet. Andreas Hopf, lektor vid institutionen för industridesign, LTH, har varit helt avgörande för den sista delen av arbetet där strukturer från naturen utnyttjas vid konstruktion av ett bord. Robert Bjärnemo och Giorgos Nikoleris vid institutionen för maskinkonstruktion har även bidragit med många idéer och kommentarer under arbetets gång.

Abstract

This Master's Thesis has shown possible applications of genetic optimization algorithms (GA) within design.

A framework for structural optimization with genetic algorithms was built and used to show the possibilities of using GA to create concept designs given just the boundary conditions and requirements of a structure.

A comparison between the optimization of the load bearing beam of an automated crane using traditional development methods and using the developed framework was then carried out and was found to give very promising results.

The final project was to adapt a very light and strong structure found in nature, which can be described by a Voronoi diagram, to a table using GA. The use of natural structures in design has long been prohibited by the complexity of the structures, which makes them very difficult to adapt in a way that is possible to manufacture, cost effective and structurally sound.

The developed computer application made it possible for a customer to define an arbitrary contour of a table, the position of the legs and its height and then have it optimized to be as light and strong as possible and still meet manufacturing requirements.

Keywords: product development, engineering design, industrial design, genetic algorithms, optimization, automated concept development, design from nature, Voronoi diagram

Sammanfattning

Problemställningen i arbetet har varit hur genetiska algoritmer, som är en typ av evolutionär algoritm, kan tillämpas inom konstruktion. GA har under de senaste åren varit föremål för intensiv forskning och har tillämpats framgångsrikt inom vitt skilda områden (Koza et al., 2005).

Genetiska algoritmer

Den första delen av arbetet undersökte GA:s uppbyggnad och verkningssätt för att kartlägga den bakomliggande tekniken och vad som krävdes för en implementering.

GA utgår från samma principer som evolutionen och börjar med en population lösningar på det uppställda problemet. Varje lösning, eller individ, beskrivs av sin "arvsmassa", eller *genom*. Varje individ testas mot de uppställda kraven och de individer som klarar sig bäst korsas med varandra och muteras för att skapa nästa generation. Denna process fortsätter sedan till ett stoppkriterium uppnått, till exempel antal generationer som förflutit. De delar som krävs för att tillämpa GA till ett problem är:

En **representation** av lösningarna där deras uppbyggnad, eller *genom* beskrivs.

En **utvärderingsfunktion** som utvärderar varje potentiell lösning och avgör om den klarar de krav som är ställda och poängsätter därefter den efter någon egenskap hos lösningen, till exempel dess vikt eller kostnad.

Även en **selektionsfunktion** som väljer ut individer till nästa generation, en **stoppfunktion** som avgör när sökningen skall avslutas, och en allmän **styrfunktion** som driver hela processen måste skapas, men de ser i allmänhet ut på samma sätt oavsett vilket problem som undersöks.

När algoritmens uppbyggnad var klarlagd analyserades tidigare utförda tillämpningar inom antenndesign till NASA:s rymdprogram (Lohn et al., 2003), turbinutformning (Jun et al., 1996), och design av nya förstärkarkretsar (Koza et al., 2005). Dessa analyser användes sedan för att avgöra vad som gjort dem lämpliga för att lösas med GA.

Problemen var komplexa och det existerade inte några etablerade metoder för att lösa dem, de hade alla tidigare lösts genom upprepade försök och ommodelleringar. Konstrukörerna fick förlita sig på erfarenhet och empiriska studier. Dock fanns det i alla tre exempel väl fungerande metoder för automatisk utvärdering, och lösningarna kunde effektivt representeras av ett *genom*.

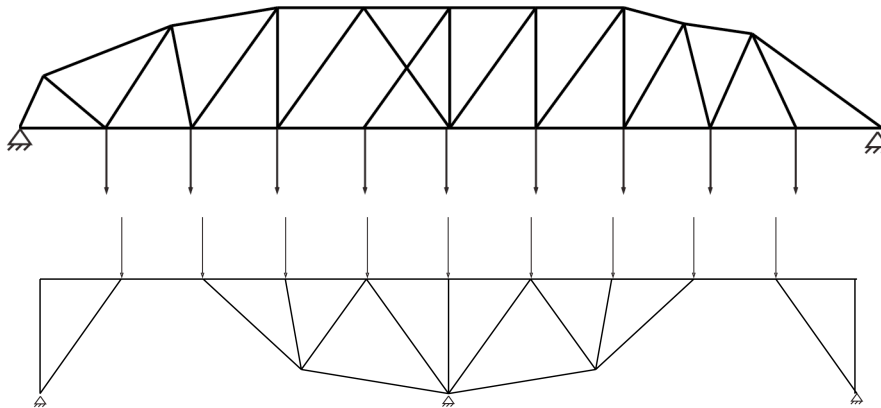
Med hjälp av dessa slutsatser beskrevs sedan nya tillämpningar inom konstruktion och design. Bland dessa valdes konstruktion av fackverk och ramverk och anpassning av strukturer från naturen till nya produkter som lämpliga områden.

Ett datorprogram för konstruktion och optimering av fackverk i GA implementerades efter att olika tekniker och programvaror utvärderats. Valet av verktyg var MATLAB (MathWorks,) med GAOT (Houck, Joines & Kay, 1995) som är en *toolbox* för genetiska algoritmer i MATLAB.

Därefter verifierades kodramverket mot problem som tidigare använts av Giger & Ermanni (2006) och Stolpe & Svanberg (2003). Resultaten stämde väl överens och överträffade i ett fall de resultat Giger & Ermanni erhållit.

Automatisk strukturoptimering

När ett fungerande ramverk hade utformats tillämpades det för att generera koncept till broar endast genom att ange randvillkor och ett antal krav på strukturen som skulle uppfyllas. Två olika randvillkor och lastfall angavs och ett antal olika koncept för varje fall skapades, se Figur 1. I båda fallen var optimeringsmålet att finna en så lätt bro som möjligt.

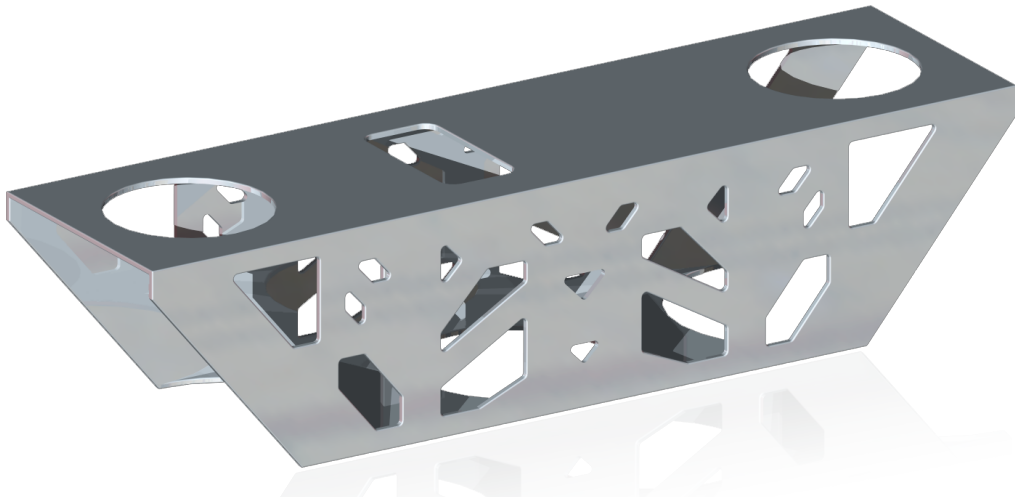


Figur 1: Exempel på två genererade brostrukturer för olika lastfall

Efter konceptgenereringen utnyttjades det framtagna programmet för att optimera en automatisk lyftkrans bärande balk. Balkens vikt skulle vara så låg som möjligt medan dess förvriddning på grund av sin last hölls inom en given gräns.

En jämförelse gjordes med en idag vanlig teknik– att stegvis optimera strukturen manuellt genom att med hjälp av ett strukturanalysprogram beräkna var material kan tas bort, och sedan modellera om konstruktionen efter dessa resultat. Detta upprepas till en konstruktion som uppfyller de satta målen uppnås.

Resultatet av optimeringen med GA var en viktminskning på 23 %, lägre förvriddning av balken och lägre maximala spänning i strukturen jämfört den traditionellt utformade balken. Se Figur 2.



Figur 2: Optimerad balk

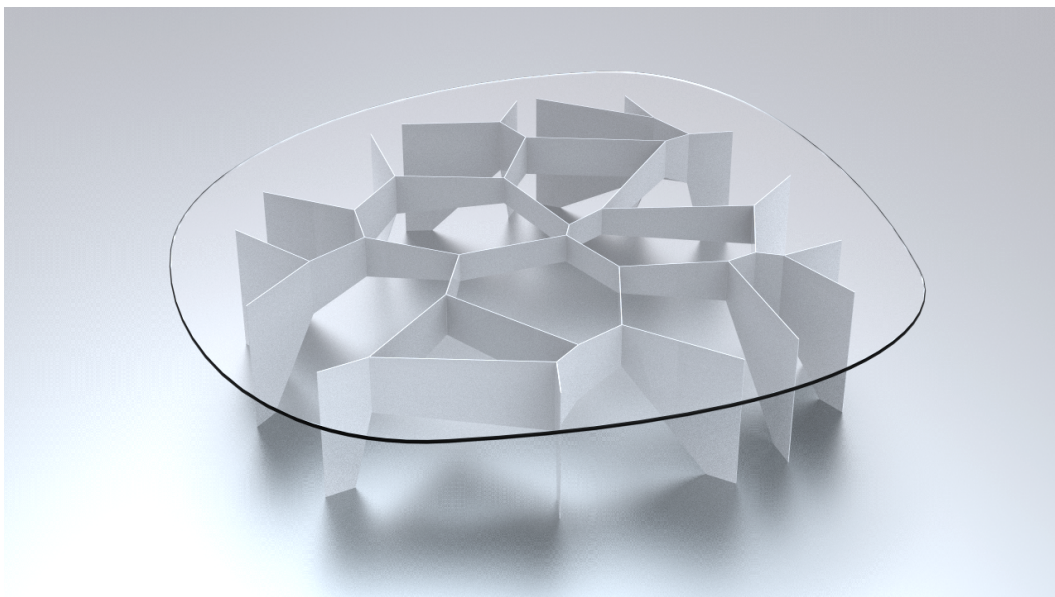
Naturinspirerad design

Sista delen av arbetet utfördes tillsammans med Andreas Hopf, lektor vid industridesign, LTH, som forskat kring tillämpningar för strukturer från naturen inom produktdesign.

Tidigare har användningen av strukturer från naturen varit begränsad av deras komplexitet vilket gjort dem svåra att tillverka och optimera för hållfasthet och vikt. Detta problem angreps med hjälp av det utvecklade ramverket för strukturoptimering.

Produkten som skulle utformas var ett bord uppbyggd av en voronoistruktur, vilken återfinns i naturen i starka och lätta strukturer, till exempel i bikupor (Beukers & van Hinte, 2005). Bordet skulle utformas på ett sådant sätt att det gick att tillverka, samtidigt som det skulle vara hållfast och ha så låg kostnad och vikt som möjligt.

Resultatet var ett program där en kund kunde definiera en godtycklig bordskontur och sedan erhålla ett optimerat bord, redo för tillverkning. Detta möjliggör att varje kund får ett unikt bord. Ett exempel på ett genererat bord visas i Figur 3.



Figur 3: Den slutgiltiga bordsstrukturen

Innehållsförteckning

1. Introduktion.....	1
2. Mål.....	3
3. Metod.....	5
4. Genetiska algoritmer	7
4.1 Presentation av tekniken	7
4.2 Exempel på tillämpningar för genetiska algoritmer	8
4.2.1 Utveckling av antenn för NASA:s rymdprogram (Lohn et al., 2003).....	8
4.2.2 Interaktiv konceptdesign (Parmee & Bonham, 2000).....	9
4.2.3 Automatiserad kreativitet och uppfinnande (Koza et al., 2005)	10
4.3 Analys av de olika implementeringarna	11
4.3.1 Antenndesign	11
4.3.2 Interaktiv konceptgenerering.....	11
4.3.3 Automatiserat uppfinnande	12
4.4 Slutsatser	12
4.5 Förslag på tillämpningar för GA	13
4.6 Valda områden.....	13
5. Implementering och verifiering av GA.....	15
5.1 Inledning	15
5.2 Problemformulering	15
5.3 Krav på programmet	16
5.4 Kartläggning av verktyg	16
5.4.1 Val av verktyg.....	17
5.5 Representation	17
5.6 Genetiska operatorer.....	19
5.6.1 Mutation.....	19
5.6.2 Korsning	20
5.7 Utvärderingsfunktion.....	21
5.7.1 Giltighetskontroll.....	21
5.7.2 Krav på strukturen	22
5.7.3 Poängsättning	22
5.8 Selektion	22
5.9 Sökparametrar	22
5.10 Implementering i programkod	22
5.11 Resultat.....	23
5.11.1 Optimering av endast tvärsnittsarea.....	23
5.11.2 Optimering av tvärsnittsarea och nodplacering.....	23
5.12 Diskussion	25
6. Konceptgenerering av brostrukturer	27
6.1 Problemformulering	27

6.2	Metod.....	28
6.3	Resultat.....	28
6.4	Diskussion	29
7.	Optimering av balk till lyftkran	31
7.1	Problemformulering	31
7.2	Metod.....	33
7.2.1	Anpassning av programmet	33
7.2.2	Överföring till solidmodell	34
7.3	Resultat.....	34
7.4	Diskussion	40
8.	Konstruktion av voronoistruktur	41
8.1	Inledning	41
8.2	Val av produkt	42
8.3	Tillverkningsmetod.....	43
8.4	Problemformulering	44
8.5	Implementering	44
8.5.1	Kundspecificerade parametrar	44
8.5.2	Skapandet av bordsstrukturen	44
8.5.3	Variabler för optimeringen	49
8.5.4	Representation och modellering av strukturen.....	49
8.5.5	Utvärderingsfunktion	50
8.5.6	Genetiska operatörer	51
8.5.7	Sökparametrar.....	52
8.6	Resultat.....	52
8.7	Diskussion	56
9.	Slutsatser och rekommendationer.....	59
	Referenser.....	61
	Bilaga 1.....	1
	Bilaga 2.....	1
	Bilaga 3.....	1

1. Introduktion

Konstruktion har länge varit ett område som krävt stor erfarenhet och lång utbildning. Många delar av produktutvecklingsprocessen är svåra att automatisera då de bygger på erfarenhet, kreativitet och ofta innefattar många bedömningar som är kvalitativa snarare än kvantitativa (Koza et al., 2005).

Artificiell intelligens spelar en allt större roll inom konstruktion och design, ett av de senaste tillskotten till denna utveckling är så kallade genetiska algoritmer som beskrevs av Holland (Holland, 1975). De har tidigare varit begränsade av att de kräver stora beräkningsresurser, men deras användning har ökat i takt med beräkningskraften. GA har visats vara en kraftfull metod för att finna lösningar på komplexa problem och har bland annat använts till att utforma antenner till NASA:s rymdprogram (Lohn et al., 2003), skapa nya turbiner (Jun et al., 1996), och för att uppfinna nya förstärkarkretsar med goda resultat (Koza et al., 2005).

Automatisk optimering av viktiga mått hos produkter är nu relativt vanligt inom industrin, men de verktyg som finns för optimering är begränsade till problem med relativt få variabler och gör att optimeringen oftast först sker i slutfasen av utvecklingsarbetet. GA har uppvisat en förmåga att kunna hantera större sökrum och öppnar alltså upp nya möjligheter inom konceptgenerering och strukturoptimering (Koza et al., 2005).

Genom automatisering av vissa delar av konstruktörsarbetet kan bättre produkter skapas. Detta kan ske genom att optimera en produktkostnad, genom att minimera materialanvändning, genom att snabbt kunna generera olika konceptlösningar på ett problem, eller till och med genom att produkter som tidigare varit alltför komplexa för en mänsklig konstruktör att utarbeta nu automatiskt kan utformas artificiellt.

2. Mål

Målet med detta examensarbete har varit att beskriva GA:s verkningssätt, fördelar och nackdelar och demonstrera processen för att implementera GA inom konstruktion och design. Målet har även varit att undersöka vilka tillämpningar som finns inom produktutveckling, och lägga grunden till framtida projekt inom området.

3. Metod

Projektets upplägg har varit att först kartlägga GA:s uppbyggnad, sedan undersöka några tidigare tillämpningar för att visa vilken typ av problem som lämpar sig för GA och på så sätt kunna finna nya tillämpningar inom produktutveckling, särskilt inom konstruktion och design, och med dessa visa implementeringsprocessen och möjligheterna med GA.

Första delen av rapporten undersökte GA och dess verkningsätt för att bättre skapa en uppfattning om den bakomliggande tekniken. Sedan kartlagdes tre tidigare tillämpningar av GA: utformning av antenner till NASAs rymdprogram (Lohn et al., 2003), optimering av nya turbiner (Jun et al., 1996), och konstruktion av nya förstärkarkretsar (Koza et al., 2005). Anledningarna till dessa tillämpningars lämplighet för GA analyserades sedan och med hjälp av dessa slutsatser hittades tillämpningar inom konstruktion och design.

När intressanta tillämpningar funnits, vilket beslutades vara inom strukturoptimering och design, undersöktes ett antal olika programvaror och källkodspaket för GA och strukturanalys. När verktyg som tillät kort utvecklingstid och pedagogiskt upplägg funnits implementerades GA för det aktuella området.

Ett antal viktiga delar av GA implementerades, bland annat representationen av varje lösning, utvärderingsfunktionen som avgör lösningarnas lämplighet och mutations- och korsningsfunktioner.

När själva algoritmen var klar verifieras den med hjälp av ett standardoptimeringsfall som bland andra Giger & Ermanni (2006) och Stolpe & Svanberg (2003) använt. Problemet var att optimera ett enkelt fackverks vikt genom att variera länkarnas tvärsnittsareor och position.

Därefter undersöktes möjligheten att generera koncept till broar, endast genom att ange randvillkor och ett antal kriterium som skulle uppfyllas och optimeras. Eftersom problemställningen var mycket lik den första behövdes inga förändringar i kodramverket göras.

Den framtagna tekniken utnyttjades sedan för att optimera en bärande balk till en automatisk kran i ett senare skede av utvecklingscykeln, och resultatet jämfördes med en balk som en konstruktör kommit fram till med hjälp av ett kommersiellt strukturanalysprogram och rådande optimeringsmetoder.

I sista delen av projektet anpassades en stark och lätt struktur från naturen som kan beskrivas av Voronoidiagram (Dirichlet, 1850) till ett bord med hjälp av GA. Målet var att göra denna process så enkel och flexibel att en potentiell kund enkelt kunde definiera en bordskontur och automatiskt få ett bord redo för tillverkning optimerat för minimal kostnad och maximal hållfasthet.

4. Genetiska algoritmer

4.1 Presentation av tekniken

Genetiska algoritmer är en optimeringsteknik som påminner om evolutionen, där starkast överlever och fortplantar sig.

Processen inleds med att ett antal individer skapas med slumpmässigt valda egenskaper. Varje individ beskrivs av sitt *genom*.

Individerna utvärderas sedan enligt några fördefinierade kriterier, och de individer som klarar sig bäst används för att skapa nästa generation. Fortplantningen kan ske genom mutation, parning, reproduktion eller andra genetiska operationer.

Denna process upprepas sedan till en tillfredsställande individ har framkommit, en viss tid gått eller efter ett visst antal generationer förlöpt.

Processen kan ses som ett antal steg:

1. Skapa en slumpmässig population
2. **Upprepa:**
3. Utvärdera varje individs styrka enligt de krav som ställts på lösningen
4. Välj ut de bästa individerna
5. Skapa en ny population från de utvalda individerna genom olika genetiska operationer, t ex mutation eller korsning
6. **Till** en godtagbar lösning hittats eller en viss tid förlöpt
7. Processens resultat är den bästa individen som hittats

Processen är, precis som i naturen, stokastisk och resultatet ändras alltså varje gång sökningen utförs.

För att implementera GA krävs ett antal val:

Representation av de potentiella lösningarna som enkelt går att beskriva och manipulera digitalt. Detta kan vara något så enkelt som en lista med de parametrar som skall optimeras i designen, eller en "ritning" av hur själva lösningen är uppbyggd.

Utvärderingsfunktion som poängsätter varje potentiell lösning efter hur väl den klarar de utsatta målen. Här används ofta simulatorer som till exempel gör FE-beräkningar och undersöker lösningens hållfasthet eller aerodynamiska egenskaper.

Selektionsfunktion som avgör hur individerna väljs ut till nästa generation.

Sökparametrar som till exempel populationsstorlek, antal generationer och sannolikheten för olika genetiska operationer.

Slutvillkor som avgör när algoritmen kan sluta leta.

Det som främst särskiljer GA från andra metoder är dess förmåga att korsa olika individer med varandra för att skapa nya lösningar. Normalt arbetar sökalgoritmer genom att endast mutera en individ i taget (Russell & Norvig, 1995: 118-119). Anledningen till att GA är en effektiv algoritm för problem med stor sökrymd och icke-linjära samband mellan variablerna är inte helt klarlagd då metoden är svår att analysera matematiskt (Russell & Norvig, 1995). Dock är alla de effektiva och anpassade lösningarna som finns i naturen tecken på att GA är en kraftfull teknik.

4.2 Exempel på tillämpningar för genetiska algoritmer

Genetiska algoritmer är mycket anpassbara och har som resultat använts inom ett stort antal fält de senaste 20 åren. Försök har gjorts inom allt från reglerteknik till fotbollsspel; till och med inom konst (Poli, Langdon & McPhee, 2008). Några framgångsrika implementeringar inom produktutveckling tas upp här.

Tillämpningarna har tagits från journalen *Artificial Intelligence for Engineering Design*, journalen *Analysis and Manufacturing, Structural and Multidisciplinary Optimization*, och konferensen *NASA/DoD Conference on Evolvable Hardware*.

Man kan använda GA i många olika faser inom utvecklingsprocessen, här beskrivs ett försök inom konceptframtagning, ett inom optimering och ett som täcker hela konstruktionscykeln.

Val av representation av lösningar, utvärderingsfunktion och sökparametrar för algoritmerna kommer att beskrivas i varje exempel, för att sedan analyseras för att finna gemensamma faktorer som har gjort att tillämpningarna blivit framgångsrika.

4.2.1 Utveckling av antenn för NASA:s rymdprogram (Lohn et al., 2003)

Lohn et al. tog, med hjälp av GA, fram nya antenndesigner som kom att användas på NASA:s satelliter. Behovet av nya sätt att utveckla antenner hade länge varit stort eftersom processen är lång och invecklad, och inga etablerade metoder fanns för att skapa antenner med givna specifikationer. Samverkan mellan olika delar av antennen och dess omgivning var komplex och svår att överblicka. Dock fanns det goda möjligheter att simulera olika antenners prestanda.

NASA hade angivit ett antal målspecifikationer för antennen som var svåra att uppnå för mänskliga konstruktörer, och simulation av antennen och rymdfarkosten var tidskrävande. Lohn et al. bestämde sig för att använda genetiska algoritmer som tidigare visat sig användbara för att lösa komplexa problem.

4.2.1.1 Implementering av GA

Representation

Lohn et al. bestämde sig för att utgå från en tråddesign på antennen. På detta sätt kunde antenndesignen representeras genom ett antal "byggoperationer" som beskrev trådens väg:

- Framåt(längd, tjocklek)

- Roterar-x(vinkel)
- Roterar-y(vinkel)
- Roterar-z(vinkel)

Dessa operationer blev alltså antennens *genom*, som sedan manipulerades med genetiska operationer.

Utvärdering

För att utvärdera varje antenndesign undersöktes hur väl konstruktionen förhöll sig till *NASA:s* målspecifikationer. Detta gjordes med hjälp av *NASA:s* egenutvecklade antenssimulator. Ju bättre värden desto högre poäng fick konstruktionen.

Varje antenndesign utvärderades ett antal gånger med små variationer i konstruktionen för att simulera att tillverkningsprocessen inte var perfekt. Konstruktionen poängsattes sedan efter de sämsta resultaten från utvärderingen.

I början av utvecklingen användes förenklade modeller av antennens omgivning för simulationen för att påskynda beräkningarna. När ett antal lovande lösningar hade tagits fram prövades dessa i en mer realistisk, men även mer tidskrävande simulation.

Algoritmparametrar

Till en början skapades 200 slumpmässiga individer som muterades och parades under 100 generationer. Individerna muterades i 1 % av fallen, och 50 % av individerna behölls mellan generationerna.

4.2.1.2 Resultat

Efter att två lösningar erhållits byggdes dessa och testades på *NASA*. Båda visade sig klara kraven och hade dessutom ett antal fördelar gentemot den traditionella designen. Antennen hade till exempel bättre mottagning i ett större område vilket sparar ström, den var mindre invecklad att tillverka, och tog kortare tid att utveckla.

4.2.2 Interaktiv konceptdesign (Parmee & Bonham, 2000)

Parmee & Bonham använde GA för att snabbt kunna söka genom ett stort antal designlösningar, i en så kallad designrymd, *design space*. Denna snabba sökning användes för att hjälpa designers interaktivt hitta intressanta lösningar på ett givet problem. Designern kunde ändra parametrar, välja bort individer och införa nya, allt eftersom sökningen fortgick. Detta för att kombinera datorns sökförmåga med designers erfarenhet och intuition.

Försök med denna teknik gjordes på British Aerospace för att undersöka om det kunde vara ett värdefullt hjälpmedel vid utveckling av nya flygplan. Utvecklingen är en invecklad process som sträcker sig över tiotals år, och som ofta inte har tydliga designmål på grund av den utdragna designprocessen. Det finns redan etablerade lösningar och ramar för nya designer, men intresset är stort för att hitta nya, innovativa lösningar för att kunna konkurrera med andra länder.

4.2.2.1 Implementering

Projektets mål var att snabbt, och interaktivt kunna finna intressanta områden i den omfattande designrymden. Parmee & Bonham valde ett relativt komplext problem som gick ut på att undersöka hur olika parametrar påverkade ett flygplans funktion. De undersökte till exempel hur vingtjocklek och vingvinkel påverkade flygplanets prestanda.

Representation

Representationen av de olika individerna var åtta olika parametrar för flygplanskonstruktionen, till exempel noslängd, vingbredd och vingarnas vinkel i förhållande till kroppen.

Utvärdering

För att utvärdera varje lösning användes en simulator som utvecklats för att testa flygplansdesigner på British Aerospace. Elva olika nyckelvärden var grunden för den testade designens poäng. Bland dessa ingick bland annat bränsleförbrukning, manövrerbarhet och högsta hastighet.

Algoritmparametrar

För att snabbt utvärdera den stora designrymden användes ett anpassbart filter som valde ut vilka lösningar som fick skapa nästa generation. Filtret var relativt tillåtande i början för att så många olika lösningar som möjligt skulle ges en chans att utvecklas. För att sedan identifiera högpresterande områden i designrymden skärptes filtret gradvis.

Användaren kunde allteftersom ändra gränserna för parametrarna för att kunna se vad som hände med designrymden. På tre minuter kunde 250 generationer undersökas, vilket var snabbt nog för att vara rimligtvis interaktivt.

4.2.2.2 Resultat

Designern kunde snabbt se vilka lösningar som var intressanta, vilka delmål de uppfyllde och hur ändringar i kraven påverkade designen. Det visade sig lätt att undersöka lösningar utanför den traditionella designrymden. Dessutom kunde enkelt krav på designen som var orimliga, eller omöjliga att kombinera med de andra kraven identifieras. På detta sätt kunde överdrivet strikta krav identifieras.

4.2.3 Automatiserad kreativitet och uppfinnande (Koza et al., 2005)

Koza et al. gör här ett försök att artificiellt återuppfinna många av 1900-talets upptäckter inom analoga kretsar. Detta sker genom att använda genetiska algoritmer som stegvis bygger upp en krets med de önskade egenskaperna.

Valet av genetiska algoritmer motiveras med deras förmåga att efterlikna den mänskliga kreativiteten. Enligt Koza et al. är många av de största uppfinningarna resultat av att uppfinnaren brutit mot de etablerade tankemönstren och skapat något som inte logiskt kunnat härledas från tidigare resultat. Detta påminner om sättet som genetiska algoritmer fungerar på. De bygger inte på någon tidigare kunskap om lösningens form, utan är en helt stokastisk process. Detta särskiljer dem från vanliga, kunskapsbaserade algoritmer som är deterministiska och alltid ger samma lösning.

4.2.3.1 Implementering

Koza et al. valde att försöka återuppfinna den negativa återkopplingskretsen som uppfanns på 1920-talet och revolutionerade stora delar av elektronikbranschen (Black, 1977).

Målet med kretsen från början var att förstärka en signal med så låg distorsion som möjligt, till ett lågt pris.

Representation

Alla kretsar som skapas utgår från en enkel krets som sedan modifieras med hjälp av ett antal operatörer som till exempel tar bort och lägger till nya kopplingar eller lägger till nya komponenter. Representationen blir alltså dessa kretsoperationer, och i vilken ordning de utförs.

Utvärdering

Utvärderingen sker genom enkla ekvationer för kretsars förstärkning och distorsion. Varje design poängsätts efter hur nära målspecifikationen den kommer. Priset på kretsen kan enkelt uppskattas genom priset på varje komponent på 1920-talet.

Algoritmparametrar

Relativt få generationer, men desto fler individer i populationerna användes för att få fram en tillfredställande lösning.

4.2.3.2 Resultat

Efter 48 generationer har algoritmen konvergerat kring en lösning som uppfyllde de krav som ställts. Denna lösning visade sig använda sig av negativ återkoppling, precis som originaluppfinningen.

Det anmärkningsvärda är att denna lösning kommit fram utan någon djupare kunskap om kretsdesign. Den enda kunskap som tillförts är de enkla ekvationerna för förstärkning och distorsion, samt operationer för hur en krets byggs upp. Uppfinnaren av originalkretsen ägnade många års arbete åt problemet innan han nådde en lösning (Black, 1977).

4.3 Analys av de olika implementeringarna

4.3.1 Antenndesign

Antenndesign är ett invecklat problem där det är svårt att förutse vad en ändring kan ha för konsekvenser utan simulation. Det finns många olika variabler att ta hänsyn till och många specifikationer som skall uppfyllas. Det är därför lätt för andra söktekniker att fastna i lokala optima. Dock lyckades GA med hjälp av en effektiv representation och utvärderingsfunktion finna en tillfredställande lösning.

4.3.2 Interaktiv konceptgenerering

GA visar sig vara ett effektivt sätt att söka genom en komplex designrymd och finna intressanta områden att undersöka närmare. Även här är representationen av varje individ enkel och väl lämpad för genetisk manipulation i form av de åtta olika måtten på

flygplanskroppen. Tack vare tidigare utvecklade simuleringsverktyg för flygplanen är det även enkelt att utvärdera lösningarna och ställa upp klara mål på flygplanets prestanda.

4.3.3 Automatiserat uppfinnande

Effektiv representation och utvärderingsfunktion finns även i detta exempel. Här visas även att GA kan användas på problem utan någon kunskap om lösningens utseende. Det enda som krävdes för att erhålla en fungerande lösning var ett antal kretsoperationer som kunde skapa alla kretsar och en pålitlig utvärderingsfunktion.

4.4 Slutsatser

De främsta fördelarna med genetiska algoritmer är deras anpassbarhet. Användaren behöver inte känna till strukturen på lösningen i förväg. Det enda som behöver anges är kraven som ställs på lösningen. Tekniken lämpar sig alltså för problem där lösningen kan finnas inom en stor domän, och bero av många olika variabler. Förmågan att utforska stora söksomåner gör även att tekniken sällan fastnar i lokala optima utan kan hitta den bästa lösningen globalt (Koza et al., 2005).

Det som krävs är pålitliga sätt att utvärdera de potentiella lösningarna. Detta kan till exempel vara en strukturanalys eller andra beräkningar av vikt eller kostnader. GA kan även vara användbart för att hitta förbättringar på redan funna lösningar. Detta kan vara viktigt inom områden där små förbättringar kan innebära stora strategiska fördelar.

Nackdelarna är främst att processen är resurskrävande och det kan ta lång tid att komma fram till en fullgod lösning. I takt med att beräkningsstyrkan i dagens datorer ökar så blir detta dock ett allt mindre problem. Lösningarna är för det mesta approximativa, och det är sällan möjligt att bevisa att den bästa lösningen har hittats (Koza et al., 2005).

Problemen som analyseras har alla varit komplexa i sin natur, och därmed svåra för både människa och maskin att lösa enligt en specifik lösningsmetod. De har tidigare lösts med hjälp av en konstruktörs erfarenhet och kreativitet, kombinerat med maskinens beräkningsstyrka. En stor del av utvecklingsarbetet har tidigare bestått i att testa olika alternativ och på så sätt stegvis arbeta fram en lösning.

I alla de tre analyserade exemplen har representationer skapats som möjliggör effektiv manipulering och utvärdering av individerna. Även utvärderingsfunktioner som pålitligt kan simulera verkligheten på ett automatiserat och effektivt sätt har hittats.

Problem som lämpar sig för bearbetning med genetiska algoritmer verkar enligt de analyserade exemplen vara sådana som traditionellt krävt mänsklig kreativitet och varit så pass komplexa att lösningarna stegvis fått utarbetas genom upprepade tester. Problem där ännu inget generaliserat sätt att finna en lösning utvecklats. Dessa problem är de som typiskt återfinns inom konstruktion och design.

Sammanfattningsvis kan ett antal gemensamma faktorer hittas hos problemen; de är:

- Komplexa och beror på många variabler
- Svåra att lösa utan mänsklig uppfinningsrikedom
- Möjliga att simulera och utvärdera realistiskt och effektivt

- Enkla att översätta till en utvärderingsfunktion
- Sådana att lösningarna kan uttryckas i representationer som möjliggör effektiv manipulering och utvärdering

Områden där GA inte lämpar sig är där specialiserade lösningar hittats, eller problem som inte kräver någon kreativitet eller nytänkande, utan bara beräkningsstyrka. Det är även svårt att använda GA inom områden där det är svårt för en maskin att bedöma en lösnings riktighet. Detta kan till exempel röra sig om estetiska kvaliteter eller problem som är svåra att sammanfatta i en enkel representation.

Här kan dock en mänsklig operatör integreras som del av utvärderingen, vilket gjordes i det interaktiva konceptgenererandet.

4.5 Förslag på tillämpningar för GA

Ett antal områden av den karaktären som listades i 4.4 följer här.

- Alla klassiska dimensioneringsproblem, t ex inom transmissioner
- Utformning av mekanismer
- Områden där FEM kan användas för beräkningen t ex
 - Fackverk/ramverk
 - Förpackningsutformning
 - Proteser
 - Vind-, våg- och vattenkraftverk
 - Värmeväxlare
- Som idégenerator där utvärderingen sker av en mänsklig operatör, möjligtvis i kombination med en digital utvärdering

4.6 Valda områden

Fackverkskonstruktioner valdes som ett inledande område då mycket forskning gjorts inom optimering av dem, och de är relativt enkla att simulera jämfört med till exempel värmeväxlare och turbinblad som kräver flödessimuleringar för att kunna optimeras. Fackverk och ramverk återfinns dessutom inom många områden, till exempel broar, kranar och möbler vilket gör att åtskilliga områden kan undersökas.

5. Implementering och verifiering av GA för det valda ändamålet

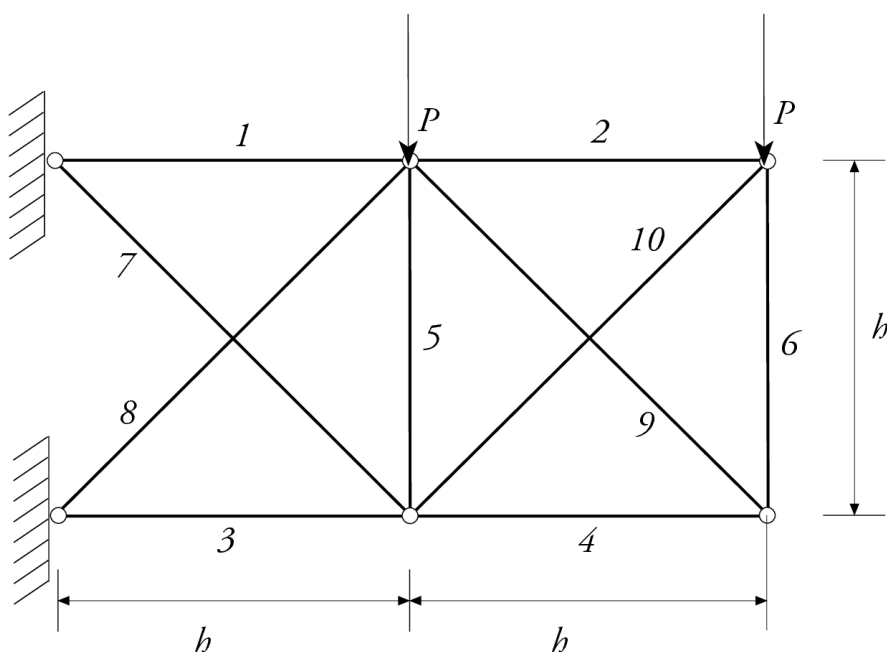
5.1 Inledning

Inledningsvis attackerades ett vanligt problem inom optimering – konstruktion och optimering av fackverk. Mycket forskning har gjorts på området och var alltså ett lämpligt exempel att verifiera GA-implementeringen med.

5.2 Problemformulering

En optimering utfördes på strukturen i Figur 4 som bland andra Stolpe & Svanberg (2003) och Giger & Ermanni (2006) har använt sig av tidigare för att testa andra optimeringsalgoritmer. Samma mått och materialdata användes för att kunna jämföra resultaten, därav avsaknaden av enheter på värdena.

Först utfördes en optimering av varje länks tvärsnittsarea, utan att förändra strukturens topologi eller nodplaceringar. Sedan optimerades både nodernas placering och länkarnas tvärsnittsareor.



Figur 4: Det gemensamma testfallet. $h = 360$, $P = 100$, $E = 10^4$, $\rho = 0.1$, $\sigma_s = 20$

5.3 Krav på programmet

De delar som var tvungna att implementeras i programstrukturen var:

- Representationen av varje individ
- Genetiska operationer
- Utvärderingsfunktion
- Selektionsfunktion

De specifikationer som skulle kunna anges för konstruktionen var:

Randvillkor Yttre krafter som verkar på strukturen, stöd som kan vara fixerade i x, y, z -led.

Optimeringsmål Den egenskap hos strukturen som skall optimeras till exempel vikt.

Krav på strukturen Egenskaper som strukturen måste uppfylla, till exempel största tillåtna spänning eller deformation.

Materialdata Data för länkarnas tvärsnitt och material.

5.4 Kartläggning av verktyg

Flera aspekter var viktiga vid val av verktyg för att implementera en fackverksoptimerare med GA. Eftersom det är en beräkningsintensiv algoritm så krävdes verktyg som hade hög prestanda och som enkelt kunde utnyttja mer datorkraft och många olika plattformar om det skulle vara nödvändigt. Det som främst tar tid i GA är utvärderingsfunktionen, vilket innebar att ett snabbt och kraftfullt FE-ramverk var nödvändigt.

Det var även viktigt att verktygen var så generella som möjligt för att de skall kunna anpassas till så många olika problem som möjligt och vara enkla att utveckla i framtiden.

För att det skulle vara enkelt att vidareutveckla verktygen var bra dokumentation och logisk programstruktur viktigt.

Många paket inom GA och FEM finns fritt tillgängliga som *open source* i C/C++. Exempel är *Paradiseo* (Cahon, Melab & Talbi, 2004) och *libMesh* (Kirk et al., 2006) som båda är effektiva ramverk för genetiska algoritmer respektive FEM i C++. De är även mycket anpassbara och en hel del dokumentation finns att tillgå. Dock är de svåra och tidskrävande att installera och anpassa för många olika processorarkitekturer och operativsystem. De har även en ganska brant inlärningskurva och användaren måste vara van vid att arbeta med C++ programmering. Detta gör att de inte lämpar sig för mindre projekt utan snarare för stora implementeringar där erfarna programmerare kan assistera vid utvecklingen.

Ett programpaket som kombinerar snabba beräkningar med ett enkelt och intuitivt programmeringsspråk är *MATLAB*:s skriptspråk (MathWorks,). Här finns möjlighet att utnyttja tusentals färdiga och optimerade beräkningsfunktioner och programpaket från tredjepartsutvecklare. Detta gör förkunskapskraven så låga som möjligt och underlättar

för personer utan omfattande programmeringsfarenhet. *MATLAB* utvecklas dessutom hela tiden och har stöd för alla större plattformar samt klusterberäkningar.

Förutom *MATLAB*:s egna implementering av GA, som inte är speciellt anpassbar utan endast kan optimera numeriska parametrar, finns en del färdiga implementeringar att tillgå. Av dessa var *The Genetic Algorithm Optimization Toolbox (GAOT)* (Houck, Joines & Kay, 1995) ett mycket anpassbart och relativt effektivt alternativ.

MATLAB:s inbyggda kommandon gör det även lätt att grafiskt representera lösningsförloppet och lösningar med hjälp av grafer och automatiskt genererade figurer.

För själva FEM-beräkningarna finns *COMSOL* (COMSOL, n.d.) som är en *toolbox* till *MATLAB* som klarar av avancerade FEM-beräkningar inom vitt skilda områden. Det går att relativt enkelt styra från *MATLAB*.

Från LTH finns även *CALFEM* (Austrell et al., 2004) som är en snabb och enkel implementering som passar bra för enklare beräkningar av till exempel fackverk och ramverk. Paketet är skapat i undervisningssyfte och är välkommenterat och pedagogiskt upplagt.

5.4.1 Val av verktyg

Målet för arbetet var inte att skapa ett kommersiellt gångbart optimeringsprogram, utan snarare att demonstrera vad som är möjligt att göra med GA inom produktutveckling. Alltså var programmets hastighet inte lika viktigt som dess pedagogiska värde och möjlighet till vidare utveckling. Detta gör att *MATLAB* med sin stora mängd inbyggda funktioner och stöd för många plattformar var väl lämpat för denna uppgift. *GAOT* är pedagogiskt upplagt och enkelt att anpassa efter behov.

Vad gäller FE-analyserna är både *COMSOL* och *CALFEM* bra alternativ, men för enklare analyser av fack- och ramverk bedömdes *CALFEM* vara snabbare, mer pedagogiskt och mer överskådligt.

5.5 Representation

En viktig del av implementeringen är att välja en flexibel och snabb representation för individerna. För fackverk finns flera olika representationer beskrivna i litteraturen. Här beskrivs två som använts med goda resultat. Fler exempel på representationer finns hos Kawamura, Ohmori & Kito (2002), Kelesoglu (2007), och V. Toğan(2006)

Coello & Christiansen (2000) representerar fackverken endast med varje länks tvärsnittsarea:

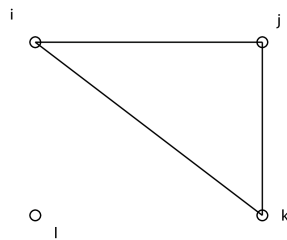
$$genom = \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{pmatrix}$$

Fackverkens topologi är alltså fixerad. Detta ger en representation som är effektiv men som kraftigt begränsar lösningsrymden. För redan definierade fackverksproblem lämpar sig detta väl, men eftersom GA:s styrka ligger i att kunna hantera komplexa problem med många variabler kan detta vara en allt för hämmad representation.

Giger & Ermanni (2006) använder sig av en friare representation som bygger på grafteori vilket gör att det enkelt går att modifiera strukturens topologi genom att ta bort eller lägga till länkelement eller nodpunkter. På detta sätt kan alla fackverkskonstruktioner skapas.

Grafrepresentationen innebär att strukturen beskrivs genom en koordinatvektor och en kopplingsmatris (*adjacency matrix*) som redogör för vilka noder som är kopplade till vilka. Detta gör det enkelt att lägga till och ta bort element; och avgöra hur många element som är kopplade till varje nod:

$$\text{genom} = \begin{bmatrix} & i & j & k & l \\ i & 0 & 1 & 1 & 0 \\ j & 1 & 0 & 1 & 0 \\ k & 1 & 1 & 0 & 0 \\ l & 0 & 0 & 0 & 0 \end{bmatrix}$$



Figur 5: Grafen som representeras av kopplingsmatrisen ovan

En etta i matrisen på rad i och kolumn j innebär att nod i är kopplad till nod j . Detta innebär att matrisen även kommer ha en etta på rad j och kolumn i , och är därför alltid symmetrisk. Se Figur 5.

För att ta bort ett element ersätts helt enkelt ettorna för de nodpunkterna som elementet är kopplat mellan med nollor. För att ta bort en nod ersätts alla ettorna i nodens rad och kolumn med nollor. För att underlätta för de genetiska operationerna tas alltså aldrig en nods rad och kolumn bort helt utan ersätts endast med nollor.

För att representera länkarnas tvärsnittsareor kan kopplingen mellan två noder beskrivas av länkens tvärsnittsarea istället för en etta i kopplingsmatrisen:

$$genom = \begin{bmatrix} & i & j & k & l \\ i & 0 & A_{i,j} & A_{i,k} & 0 \\ j & A_{j,i} & 0 & A_{j,k} & 0 \\ k & A_{k,i} & A_{k,j} & 0 & 0 \\ l & 0 & 0 & 0 & 0 \end{bmatrix}$$

5.6 Genetiska operatörer

5.6.1 Mutation

5.6.1.1 Mutation av länkar

De enklaste genetiska operatörerna för att ändra strukturens topologi är att lägga till och ta bort element. Detta sker som beskrivet genom att ettor tas bort eller läggs till i kopplingsmatrisen för elementets nodpunkter:

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \rightarrow A_{mut} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

För att göra det mer sannolikt att mutationerna genererar en giltig lösning kontrolleras att element inte tas bort från noder som är utsatta för randvillkor om de har färre än två kopplingar till andra noder.

Det sker även en kontroll av att nya länkar kopplas till noder som redan har andra kopplingar eller är fixerade för att undvika underbestämda strukturer.

5.6.1.2 Mutation av noder

Eftersom enbart mutation av länkar lätt skapar underbestämda konstruktioner måste även noder kunna läggas till och tas bort. Detta sker som beskrivet genom att ettorna i nodens rad och kolumn i kopplingsmatrisen ersätts med nollor:

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \rightarrow A_{mut} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Det sker även en kontroll av att inga noder utsatta för randvillkor tas bort.

En enkel muteringsalgoritm ändrar slumpmässigt positionen för nodpunkterna:

$$v = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{pmatrix} \rightarrow v_{mut} = \begin{pmatrix} x_1 + \delta_{x_1} & y_1 + \delta_{y_1} \\ x_2 + \delta_{x_2} & y_2 + \delta_{y_2} \\ x_3 + \delta_{x_3} & y_3 + \delta_{y_3} \end{pmatrix}$$

De enda punkterna som inte kan flyttas är de som är påverkade av externa krafter eller randvillkor.

Eftersom individerna blir allt bättre anpassade ju fler generationer som går så sker även färre och mindre omfattande mutationer i de senare generationerna.

5.6.2 Korsning

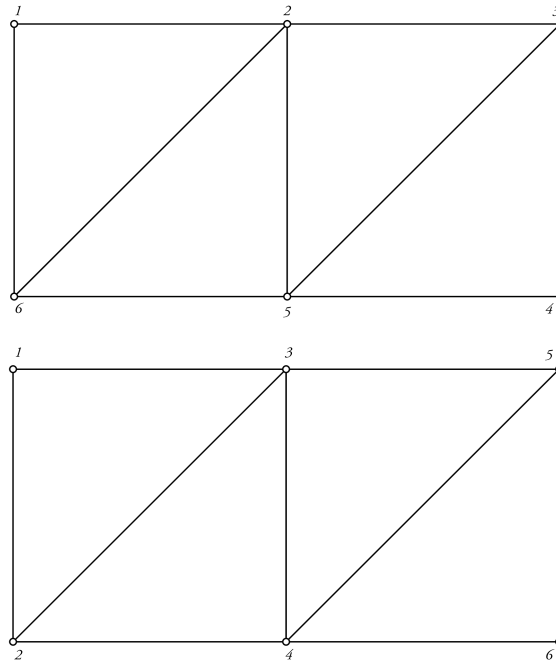
Vid korsning av två lösningar byter de delar av sina kopplingsmatriser med varandra:

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Detta fungerar bra så länge nodpunkterna alltid har samma koordinater.

Om de kan ha olika positioner i olika individer så genereras dock sällan giltig eller användbar avkomma. Detta kan lösas genom att nodnumreringen optimeras inför korsningen med hjälp av Cuthill-McKee-algoritmen (Cuthill & McKee, 1969). Syftet med algoritmen är att minimera bandbredden för en symmetrisk matris, det vill säga att sortera om matrisen på ett sådant sätt att de värden som är skilda från noll hamnar så nära diagonalen i matrisen som möjligt. Minimeras bandbredden i kopplingsmatrisen gör detta att element som är kopplade till varandra får liknande numrering, vilket i sin tur innebär att korsningen av två kopplingsmatriser har större sannolikhet att bli lyckad om de gått genom omnumreringsalgoritmen först:

$$\begin{bmatrix} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 1 & & & & 1 \\ 2 & 1 & 0 & 1 & & 1 & 1 \\ 3 & & 1 & 0 & 1 & 1 & \\ 4 & & & 1 & 0 & 1 & \\ 5 & & 1 & 1 & 1 & 0 & 1 \\ 6 & 1 & 1 & & & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 1 & 1 & & & \\ 2 & 1 & 0 & 1 & 1 & & \\ 3 & 1 & 1 & 0 & 1 & 1 & \\ 4 & & 1 & 1 & 0 & 1 & 1 \\ 5 & & & 1 & 1 & 0 & 1 \\ 6 & & & & 1 & 1 & 0 \end{bmatrix}$$



Figur 6: Demonstration av omnumrering av noder med Cuthill-McKee-algoritmen. Kopplingsmatrisen ovan beskriver nodernas kopplingar

Även koordinatvektorerna för noderna korsas:

$$P_1 = \begin{pmatrix} x_1^a & y_1^a \\ x_2^a & y_2^a \\ x_3^a & y_3^a \end{pmatrix} + P_2 = \begin{pmatrix} x_1^b & y_1^b \\ x_2^b & y_2^b \\ x_3^b & y_3^b \end{pmatrix} \rightarrow c_1 = \begin{pmatrix} x_1^a & y_1^a \\ x_2^b & y_2^b \\ x_3^b & y_3^b \end{pmatrix}, c_2 = \begin{pmatrix} x_1^b & y_1^b \\ x_2^a & y_2^a \\ x_3^a & y_3^a \end{pmatrix}$$

5.7 Utvärderingsfunktion

I utvärderingsfunktionen finns ett antal krav på strukturen, samt ett antal egenskaper som skall optimeras. I detta fall är kraven på strukturen att den inte skall överskrida den maximalt tillåtna spänningen, och att den inte skall bucklas. Egenskapen som skall optimeras är vikten.

5.7.1 Giltighetskontroll

Risken för att ogiltiga konstruktioner skapas är stor när länkar och nodpunkter skapas och tas bort vid mutering och korsning.

Detta gör att utvärderingsfunktionen måste vara kritisk till lösningarna som den får in och kontrollera deras giltighet innan den gör tidsödande beräkningar på dem.

Detta görs bland annat genom att kontrollera om lösningarna är garanterat statistiskt underbestämda, det vill säga om de inte uppfyller kravet

$$Nd-L-R \leq 0 \quad (1)$$

där N är antalet nodpunkter, d är antalet dimensioner som strukturen har, L är antalet länkelement och R är antalet frihetsgrader som är låsta på grund av randvillkoren.

Det sker även en kontroll av att alla noder som är utsatta för randvillkor har element kopplade till sig.

Delar av strukturen kan dock fortfarande vara underbestämda även om strukturen som helhet uppfyller ekvation 1. Dessa uppträcks dock när nodförskjutningarna skall beräknas då de resulterar i en olösbar ekvation.

5.7.2 Krav på strukturen

Spänningen i varje element beräknas med hjälp av *CALFEMs* metoder för att beräkna nodförskjutningar och normalkrafter i länkarna.

Spänningen i element i ges av $\sigma_i = \frac{N_i}{A_i}$, där N_i är normalkraften och A_i är länkens tvärsnittsarea.

För att undvika buckling av elementen måste tryckspänningen i elementet underskrida σ_{kr} . Om enbart cirkulära tvärsnitt används i strukturen så kan σ_{kr} beräknas enligt: $\sigma_{kr} = A_i \pi E / (2l_i)^2$ där l_i är elementets längd, och E är elasticitetsmodulen för materialet.

5.7.3 Poängsättning

Varje individs poäng sätts enligt $p = \frac{1}{w}$, där w är vikten. Vikten beräknas helt enkelt genom $w = A \cdot l \cdot \rho$, där ρ är materialets densitet. Uppfyller inte individen alla krav så får den poängen 0.

5.8 Selektion

Selektionen skedde genom att individernas sannolikhet att bli valda att fortsätta till nästa generation ökar med deras poäng.

5.9 Sökparametrar

Alla optimeringar utfördes med 50 individer och 1600 generationer för att kunna jämföra algoritmernas effektivitet med Giger & Ermanni som använde samma parametrar.

5.10 Implementering i programkod

När valen av representation, genetiska operatorer, utvärderings funktion, selektion och sökparametrar var gjorda implementerades alla funktioner i MATLAB:s skriptspråk. All källkod skrevs från grunden, utom de delar som implementerades av CALFEM, och GAOT, det vill säga bestämningen av spänningar och förskjutningar i fackverken, samt den generella sökfunktionen och selektionsfunktionen. Cuthill-McKee-algoritmen fanns redan implementerad i MATLAB. Totalt skrevs omkring 3000 rader ny kod.

5.11 Resultat

5.11.1 Optimering av endast tvärsnittsarea

Resultaten för tvärsnittsoptimeringen visas i Tabell 1. Lösningen lik de lösningar som Stolpe & Svanberg och Giger & Ermanni fått. Vikten skiljer endast 0,057 % från den Stolpe & Svanberg kom fram till.

Största spänningen i strukturen är $19,9944 < 20$ och säkerhetsmarginalen till buckling är $9,1 \times 10^{-4} > 0$.

Tabell 1: Resultatet av tvärsnittsoptimeringen

Länk	Stolpe	Giger	Nordin
1	5,00000	5,00139	5,0317
2	5,00000	5,00139	5,0218
3	70,35876	70,3595	70,3936
4	0	-	-
5	40,62165	40,6388	40,6398
6	0	-	-
7	14,14214	14,14214	14,1461
8	0	-	-
9	0	-	-
10	68,31720	68,3187	68,3343
Vikt	8553,44	8557,45	8558,3

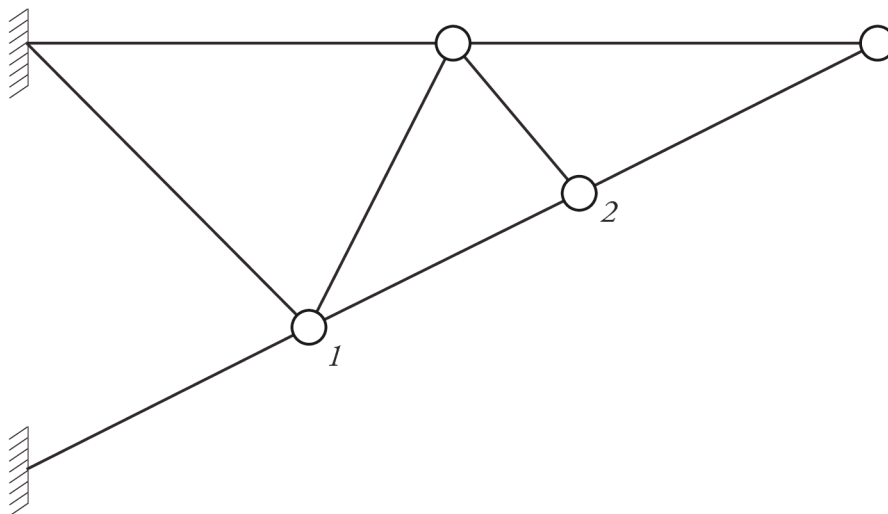
5.11.2 Optimering av tvärsnittsarea och nodplacering

Lösningen i Figur 7 erhöles efter att 50 individer utvecklats under 1600 generationer. Konstruktionens massa är 5750,2 och högsta spänning är 19,9696. Detta kan jämföras med Giger & Ermanni (2006) som erhöles lösningen i Figur 8 vilken hade massan 5898,2 och högsta spänningen 19,99. Alltså erhöles en viktminskning med 2,5 %.

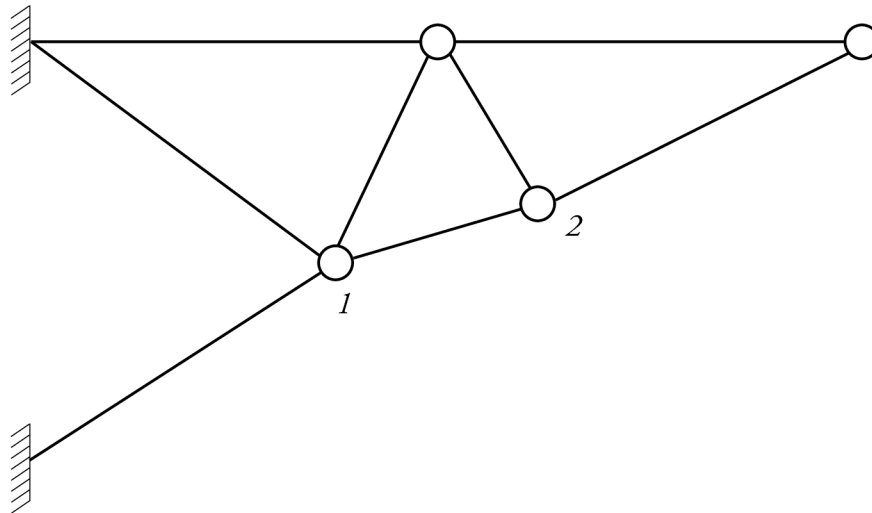
Tabell 2: Jämförelse mellan tvärsnittsareor

Länk	Giger	Nordin
1	12,8538	12,6424

2	9,48097	10,0360
3	72,6613	55,3630
4	27,0939	43,4204
5	26,4674	32,2177
6	49,7662	47,5668
7	2,53085	3,5363
8	-	-
9	2,3747	0,9483
10	-	-
Vikt	5898,2	5750,2



Figur 7: Resultat vid optimering av tvärsnittsarea och nodplacering där koordinaterna är $v_1:(239,38 \ 119,66)$ och $v_2:(467,91 \ 233,96)$



Figur 8: Resultatet som Giger & Ermanni fick där koordinaterna är $v_1:(299,62 \ 172,99)$ och $v_2:(453,59 \ 219,50)$

5.12 Diskussion

Algoritmen ger resultat som ligger nära de som tidigare erhållits med andra metoder. Vikten är något högre än den som Stolpe & Svanberg funnit, förmodligen eftersom GA är en heuristisk metod som har svårt att finna exakt optimala lösningar.

Vid optimeringen av både tvärsnitt och nodposition erhöles en lösning som var något lättare än den Giger & Ermanni funnit. Detta visar på att algoritmen är korrekt och kapabel till att skapa och optimera fackverk av detta slag. Den föreslår en lösning som fördelar areorna ganska annorlunda jämfört med Giger & Ermannis lösning vilket visar på att det undersökta problemet är öppet med många lokala optimum vilket gör det lämpat för just GA.

Det som visade sig vara viktigast för den slutgiltiga lösningens utseende var utvärderingsfunktionen. Små ändringar gjorde stora utslag i lösningarnas kvalitet. Detta gör att mest tid bör läggas på denna funktion för att säkerställa dess riktighet. Det är även i utvärderingsfunktionen som mest processortid lagts vilket ytterligare pekar på att en effektiv och korrekt utvärdering är avgörande för GA.

6. Konceptgenerering av brostrukturer

Den främsta användningen för optimeringsalgoritmer inom konstruktion har länge varit att fullända redan befintliga lösningar, med några få parametrar som kan varieras.

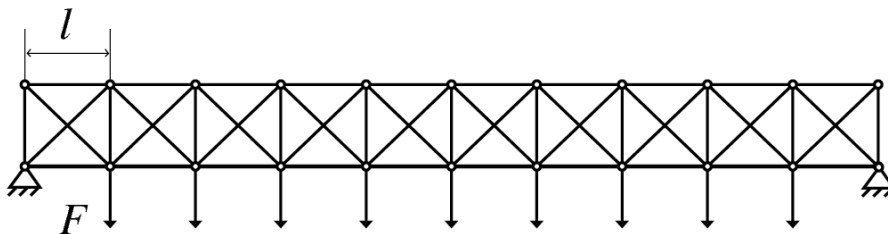
Då GA, som konstaterats i 4.4, främst kommer till sin fördel i öppna och komplicerade problem gör detta att helt andra problemtyper kan angripas. En mycket fri fas i produktutvecklingscykeln som normalt bygger på konstruktörens intuition och erfarenhet är konceptgenereringen. Här beskrivs hur det utvecklade programramverket kan utnyttjas för att snabbt skapa konceptlösningar för invecklade problem.

För att utnyttja så mycket som möjligt av tekniken som utvecklats i förra avsnittet så valdes brokonstruktioner till lämpliga testobjekt. Brodesign är även ett öppet problem att döma av den uppsjö av brodesigner som finns världen över (Janberg, 1998-2009).

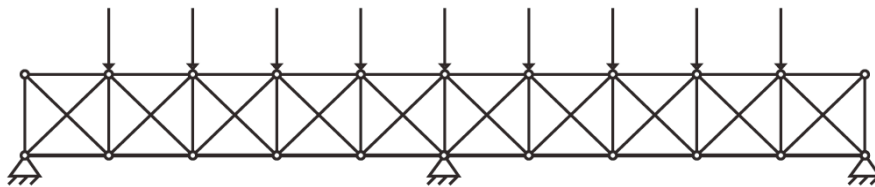
6.1 Problemformulering

Givet randvillkor och laster skall programmet generera en så lätt konstruktion som möjligt, utan att överskrida maximal drag-, tryck-, eller bucklingsspänning.

Två olika testfall formulerades. Ett med två stöd, och ett med tre stöd enligt figur 9 och figur 10 med 40 respektive 38 frihetsgrader. Fallen valdes på grund av att broar med två respektive tre stöd är vanliga och det därmed finns många verkliga konstruktioner att jämföra med.



Figur 9: Det första testfallet



Figur 10: Det andra testfallet

En kraft $F=50000\text{ N}$ är ansatt i de angivna nodpunkterna. Länkarnas tvärsnittsareor är satta till $2\times 10^{-2}\text{ m}$. Brons längd är 50 m, grundstrukturens höjd är 7 m.

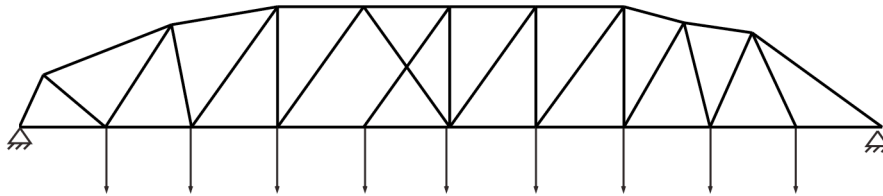
6.2 Metod

Då problemet som ställts upp var mycket likt det som ställdes upp i 5.2 så behövdes inga ändringar göras i programmet för att passa brostrukturerna.

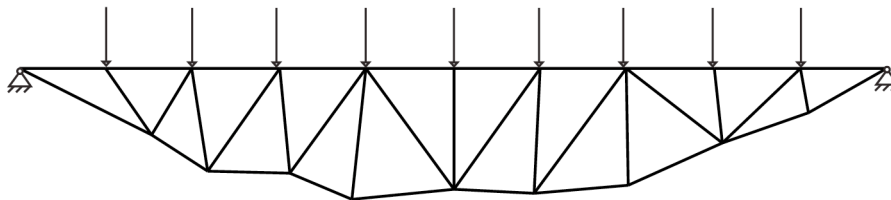
Kraven var som tidigare att flyt- och bucklingsspänningen inte fick överskridas. Optimeringsmålet var även som tidigare konstruktionens vikt.

6.3 Resultat

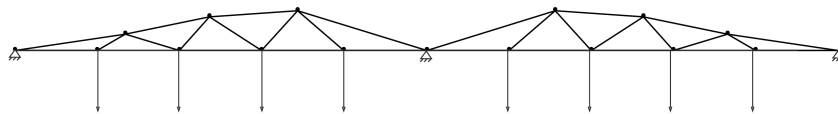
Optimeringen skedde med 50 individer under 1600 generationer, vilket tog ungefär 5 minuter per körning på en 2,2 GHz dubbelkärnig processor. I Figur 11 och Figur 12 visas lösningar på det första problemet och i Figur 13 och Figur 14 lösningar på det andra problemet.



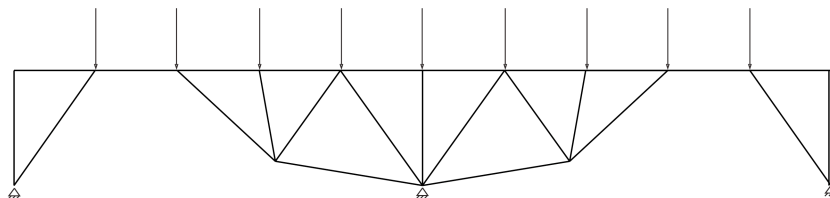
Figur 11: En lösning på det första testfallet



Figur 12: En annan lösning på det första testfallet



Figur 13: En lösning på det andra testfallet

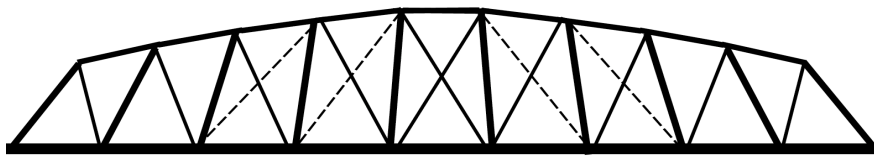


Figur 14: En annan lösning på det andra testfallet

6.4 Diskussion

Resultaten från testproblemen ligger nära redan etablerade strukturer vilket tyder på tillförlitliga lösningar. I Figur 11 syns en konstruktion som är lik en Pegrambro, vilken patenterades 1885 (Pegram, March 24, 1885). Även den andra lösningen har ett typiskt utseende för denna typ av bro. Se Figur 15 och Figur 16. Detta innebär att algoritmen lyckats återskapa strukturer som tagit mänskliga konstruktörer lång tid att komma fram till på några minuter.

Med hjälp av dessa koncept kan en konstruktör snabbt hitta intressanta lösningar och välja ut vilka som är mest lämpliga att fortsätta med. Detta gör att nya, innovativa lösningar kan skapas eftersom GA inte bygger på erfarenhet utan är helt stokastiska.



Figur 15: Pegrambro



Figur 16: Fackverksbro över Yellowstonefloden av pegramtyp

7. Optimering av balk till lyftkran

7.1 Problemformulering

För att undersöka hur väl den utvecklade algoritmen klarade av en verklig optimering valdes en konstruktion som ställde höga krav på styvhet och vikt. Resultatet jämfördes med en lösning som skapats av en konstruktör på LTH(Finnman, 2009) och optimerats med hjälp av strukturanalysprogrammet *ANSYS* (ANSYS, n.d.), se Figur 18. I *ANSYS* gjordes en topologi-optimering med hjälp av inbyggda optimeringsmetoder som visar var material kan tas bort. Dock tar inte *ANSYS* automatiskt bort material, utan konstruktören måste själv göra ändringar i solidmodellen och sedan göra en ny analys och på så sätt manuellt iterera fram en optimerad lösning. Balken som båda optimeringarna utgått från och den med traditionella verktyg optimerade balken visas i bilaga 1 respektive 2.

Balkens styvhet är viktig för dess funktion då en för stor nedböjning av motstående fästordning förhindrar den från att passa i fästpunkten i taket.

Kraven som var ställda på originalbalken var att materialets sträckgräns inte fick överskridas och förvridningen av en axel genom centrum av fästpunkten enligt Figur 24 fick inte vara mer än en 0,1 grad för att den säkert skulle passa i motsvarande fästpunkt i taket.

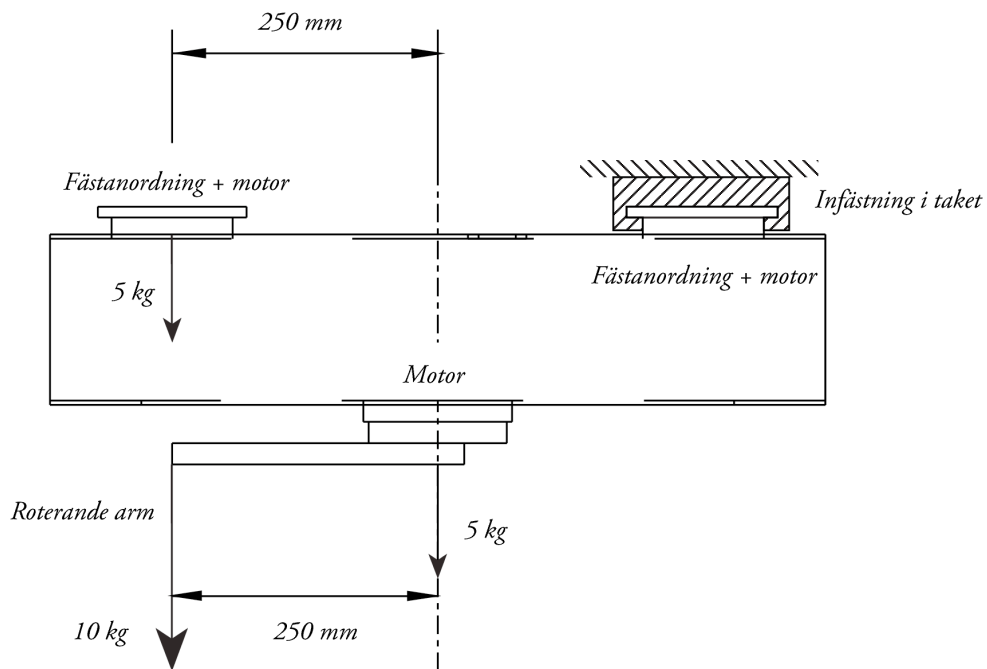
Målet är att minimera konstruktionens massa medan de krav som var ställda på originalbalken fortfarande uppfylls.

Balken skall bära upp en last på 10 kg som sitter fast i en roterande arm med längden 250 mm, se Figur 17. Konstruktionen hänger i en av de två fästordningarna i taget. Genom att rotera kring fästpunkten kan anordningen förflytta sig mellan olika fästpunkter i taket. Rotationen sker med hjälp av motorer som väger 5 kg styck.

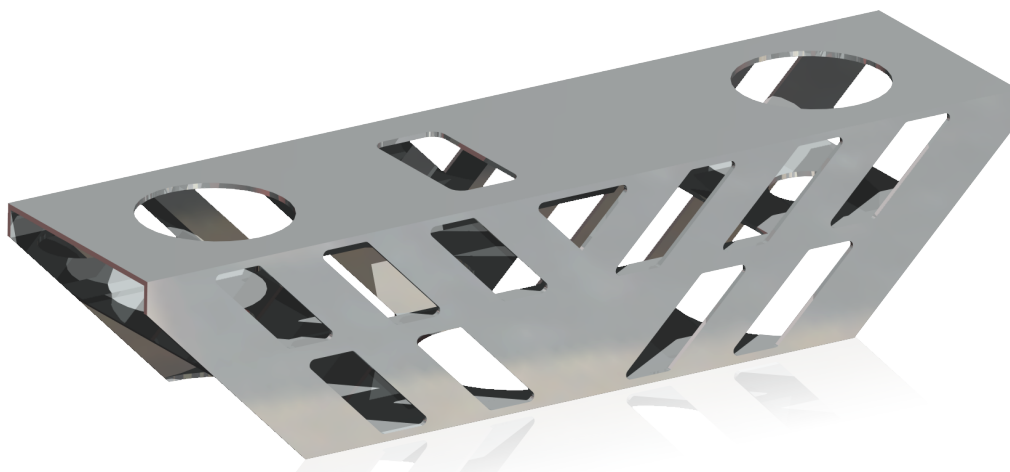
Laster och randvillkor är angivna enligt Figur 17. Materialdata är angivna i Tabell 3.

Tabell 3: Materialdata

Elasticitetsmodul	68,95 GPa
Densitet	2710 kg/m ³
Poissions ratio	0,3
Sträckgräns	280 MPa

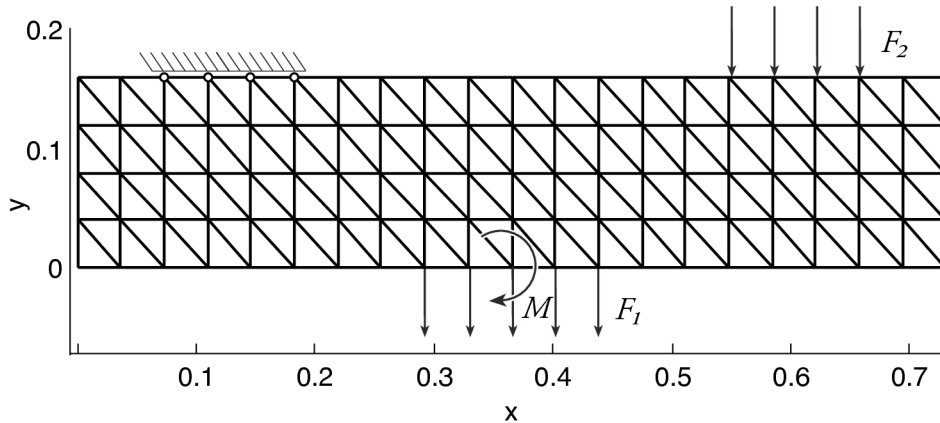


Figur 17: Den studerade balken i lastfallet som skulle optimeras



Figur 18: Den tidigare konstruerade balken

7.2 Metod



Figur 19: Utgångsstrukturen för GA, där $F_1 = 10 \cdot g N$, $F_2 = 5 \cdot g N$ och $M = 0,250 \cdot F Nm$ är fördelade över de fem noderna. Varje länk har tjockleken 4 mm och bredden 20 mm

För att optimeringen skulle gå snabbare och underlätta modelleringen av problemet gjordes optimeringen med hänsyn till enbart ett lastfall. Lastfallet valdes till det i Figur 17 då det är det lastfall som har lasten längst från fästpunkten och därmed skapar störst böjmoment, och på samma gång kan det modelleras i två dimensioner vilket gör geometrin enklare.

För att sedan verifiera den skapade balken modellerades båda balkarna med samma randvillkor i ANSYS. Här undersöktes samma lastfall som konstruktören testat mot där armen med last befinner sig i 45° vinkel mot balken.

7.2.1 Anpassning av programmet

7.2.1.1 Representation

Eftersom balkar med variabel tjocklek är komplicerat att tillverka så gavs algoritmen inte möjlighet att ändra länkarnas tvärsnitt utan endast vilka nodpunkter som var kopplade till varandra.

Istället för att som tidigare representera krafter och randvillkor med vilket nodnummer de verkar i, representerades de nu av deras angreppskoordinater.

Detta gjorde det enkelt att definiera problemets villkor utifrån en ritning eftersom randvillkoren nu inte behövde översättas till nodnummer, och gjorde även att det automatiskt gick att ändra hur fint strukturen var indelad i nodpunkter och länkar. Den slutgiltiga indelningen var enligt Figur 19.

7.2.1.2 Nya mutationsfunktioner

Koordinatrepresentationen möjliggjorde en speglingsmutation vilken speglade strukturen på mitten.

På strukturer med stort antal noder är det sällan en slumpmässig sammanlänkning av två noder ger en förbättrad struktur eftersom noderna ofta ligger långt ifrån varandra. För att avhjälpa detta ersattes det slumpmässiga nodvalet med ett där det var större sannolikhet att välja noder som ligger nära utgångsnoden.

En mutation eller snarare selektion av individer som har visat sig ge goda resultat är något som närmast kan liknas vid en naturkatastrof, där endast de bästa och mest anpassade individerna överlever. Tekniken går ut på att välja den bästa individen i en generation och sedan ersätta resten av populationen med mutationer av denna individ.

7.2.1.3 Utvärderingsfunktion

Kraven på varje individ var de enligt avsnitt 7.1.

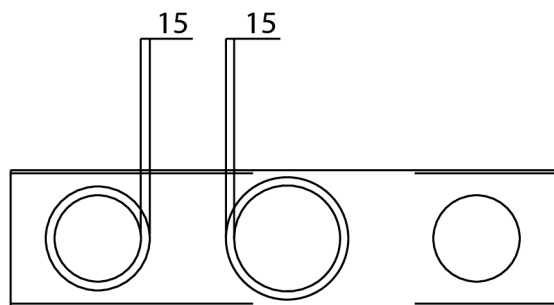
För att undvika strukturer som var extremt anpassade för just det undersökta lastfallet, men instabila för alla andra laster, infördes små slumpmässiga laster vid alla använda nodpunkter.

I övrigt var utvärderingen den samma men eftersom lasterna och randvillkoren nu var angivna i nodkoordinater fick dessa översättas till nodnummer varje gång.

7.2.2 Överföring till solidmodell

När en optimerad fackverksstruktur erhållits från programmet överfördes dess utseende till en solid balk genom att de ytor som var täckta av länkar fick vara kvar medan de kvarstående tomrummen togs bort, se Figur 21 och Figur 22.

För att simulera de skruvförband som finns i strukturen sattes lasten till att angripa ett område med 15 mm större radie än hålet för armmotorn. Fästanordningen sattes också att angripa ett område med 15 mm större radie än monteringshålet. Se Figur 20.



Figur 20: Områden för simulerade skruvförband

7.3 Resultat

Algoritmen fick arbeta under 1600 generationer med 50 individer. Detta tog ungefär en timme på en 2.2 GHz processor med två kärnor.

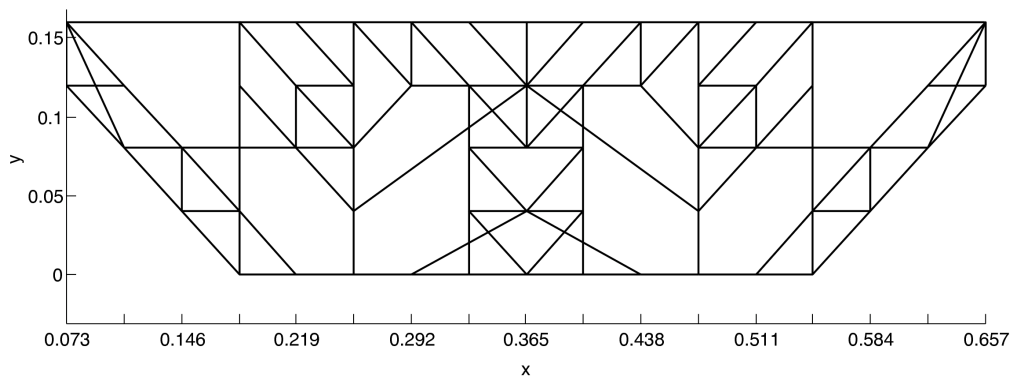
Den optimerade fackverksstrukturen visas i Figur 21, strukturen med skalenlig länktjocklek visas i Figur 22. Den resulterande balken visas i Figur 23 och bilaga 3. En

jämförelse mellan de olika balkarna i olika lastfall visas i Figur 26 till Figur 29. En bild på de laster och randvillkor som använts vid analysen visas i Figur 25.

I Tabell 4 visas vikt, högsta spänning och förvridningen av fästpunkten för de olika balkarna vid simulering i *ANSYS*.

Balken som utformats av en konstruktör har en platta i mitten av balken för att styva upp konstruktionen. Motorn som sitter fast i den övre fästpunkten här är även fästad i denna platta. Dock bör motorn inte ta upp några större krafter eller moment utan den skyddas av omgivande struktur. Plattans funktion är att motverka vridmomentet som uppkommer av lasten.

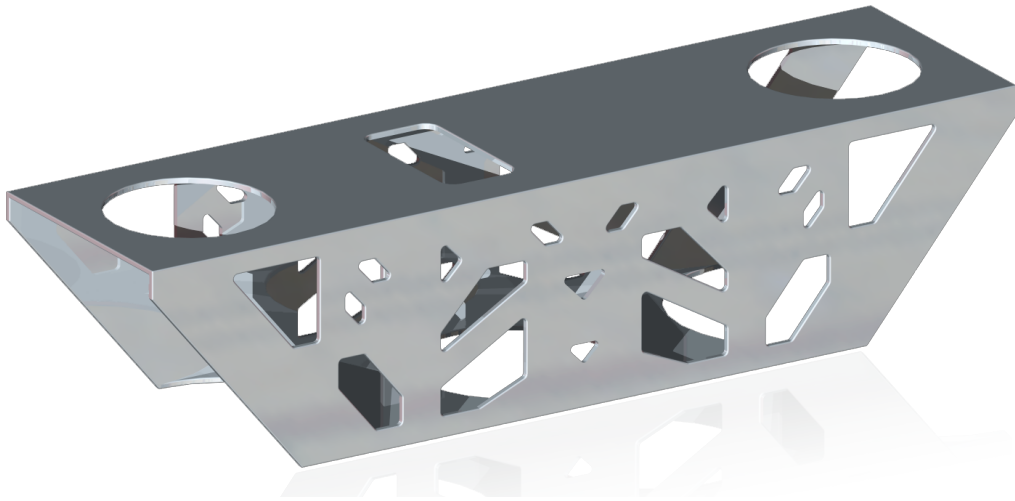
Balken utformad med GA har istället för mittplattan fått en starkare struktur längs med långsidorna och på så sätt sparar in på vikten. Motorn är här endast fästad i den övre plattan men omges av mer material för att stödja den.



Figur 21: Optimerad fackverksstruktur



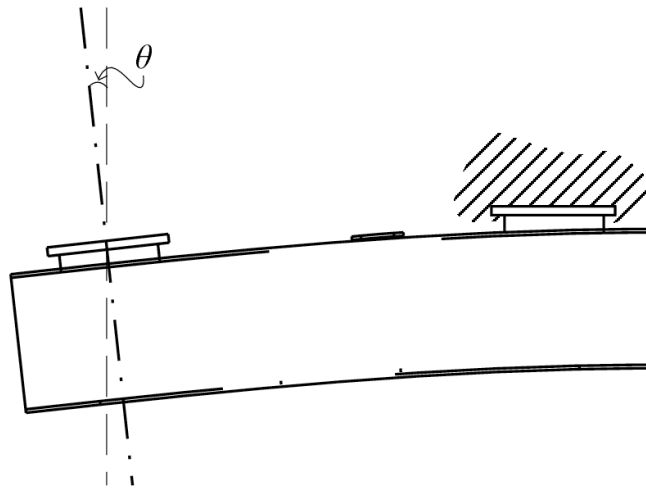
Figur 22: Optimerad fackverksstruktur med skalenlig länktjocklek



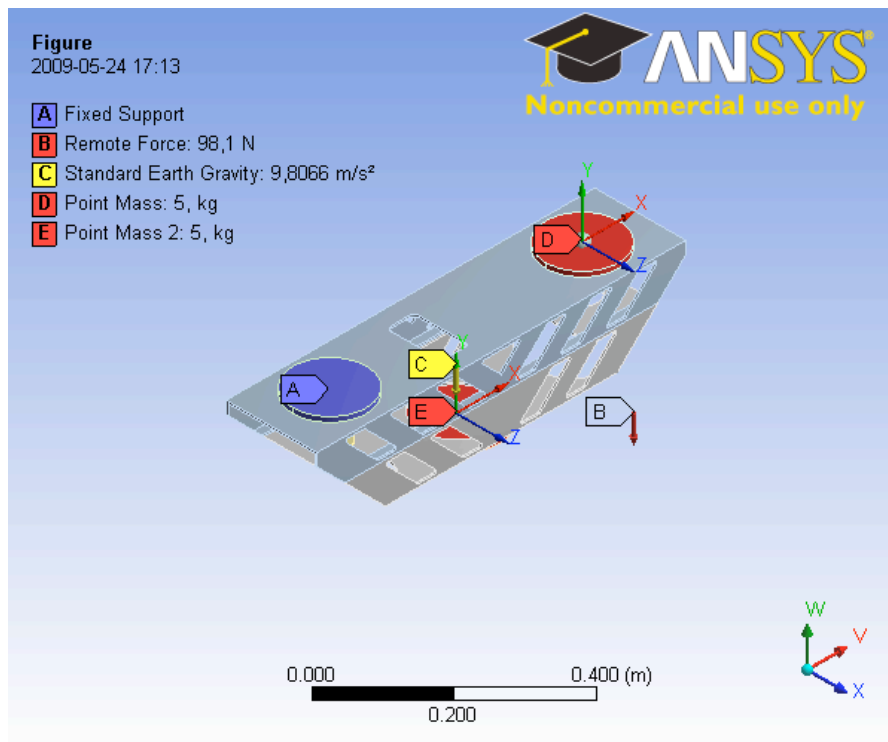
Figur 23: Optimerad solidmodell

Tabell 4: Resultat för de olika balkarna, där θ är förvriddningen i mätpunkt ett i grader, σ_{\max} den största uppkomna spänningen och m strukturens massa

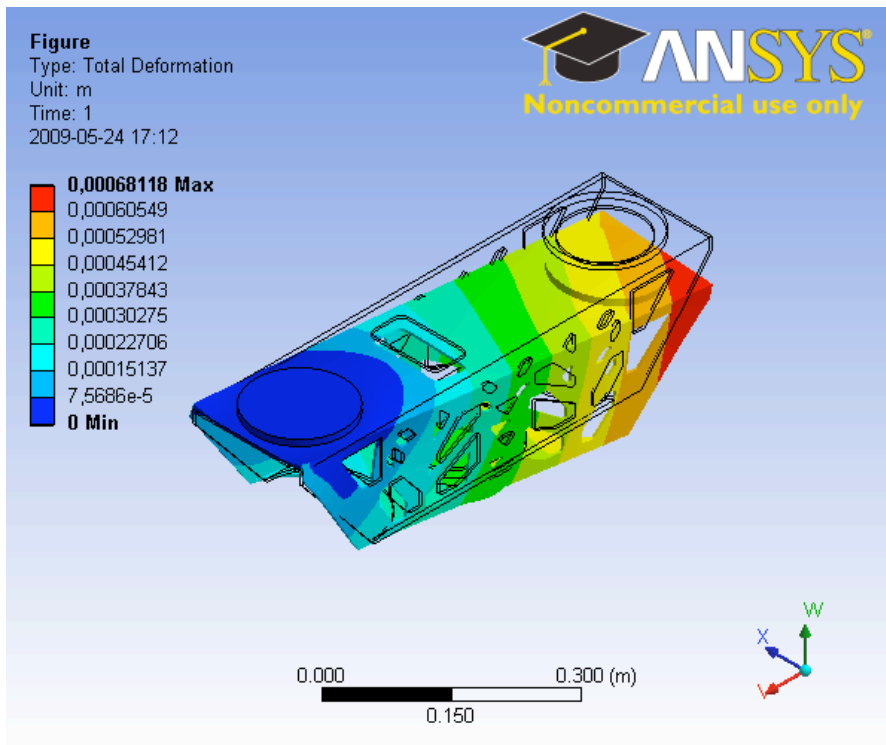
	Original	Konstruktör m h a ANSYS	GA
θ	0,051593	0,084686	0,080998
σ_{\max}	15,110	28,680	23,669
m	4,6504	3,8678	2,98



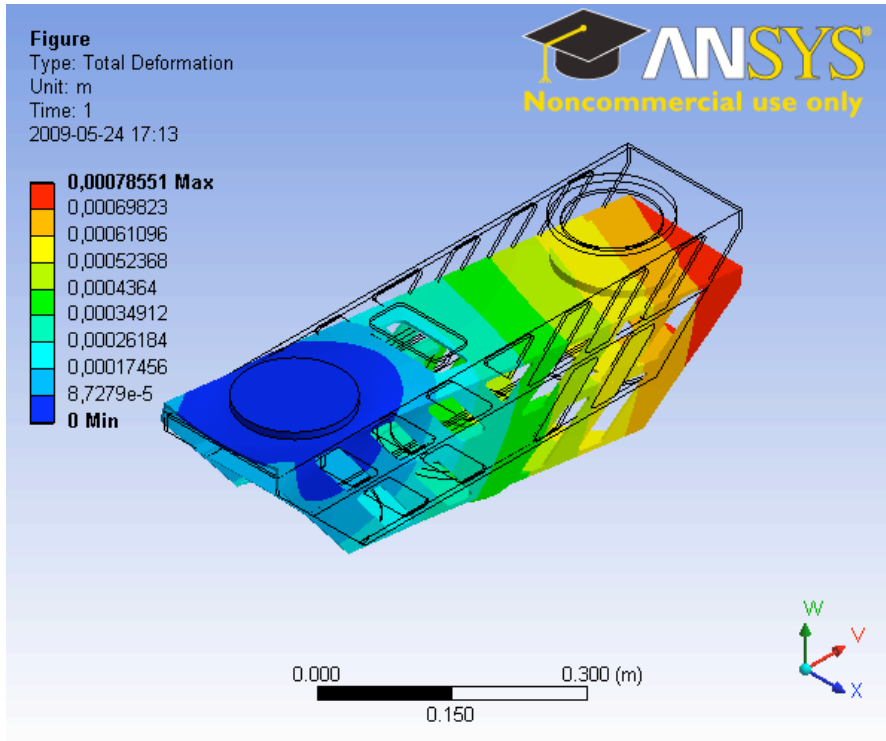
Figur 24: Förvriddningen θ sett i $x - y$ -planet



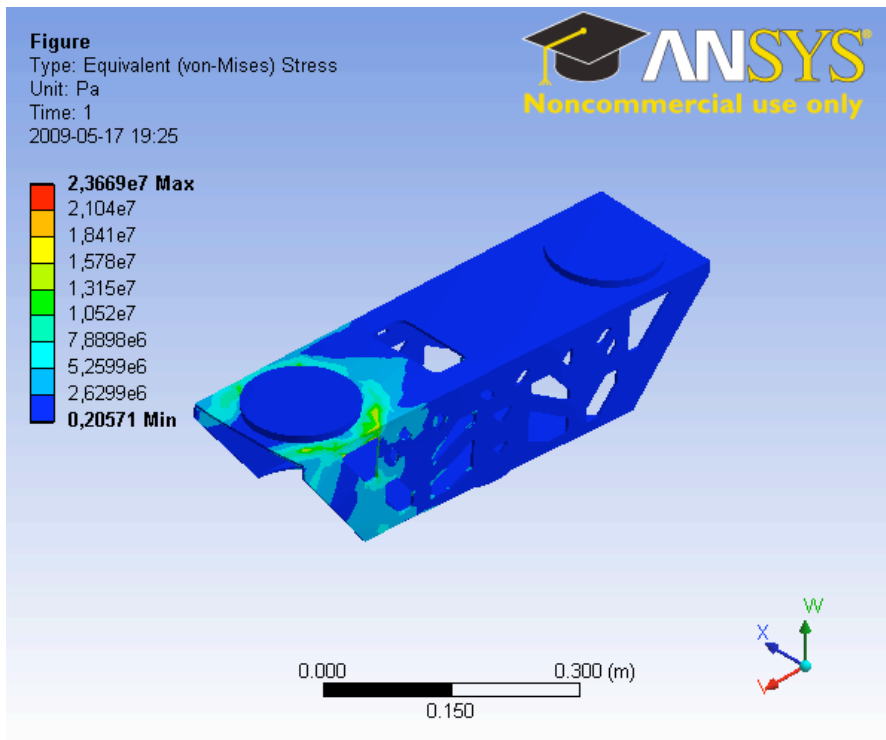
Figur 25: Laster och randvillkor för femmodelleringen



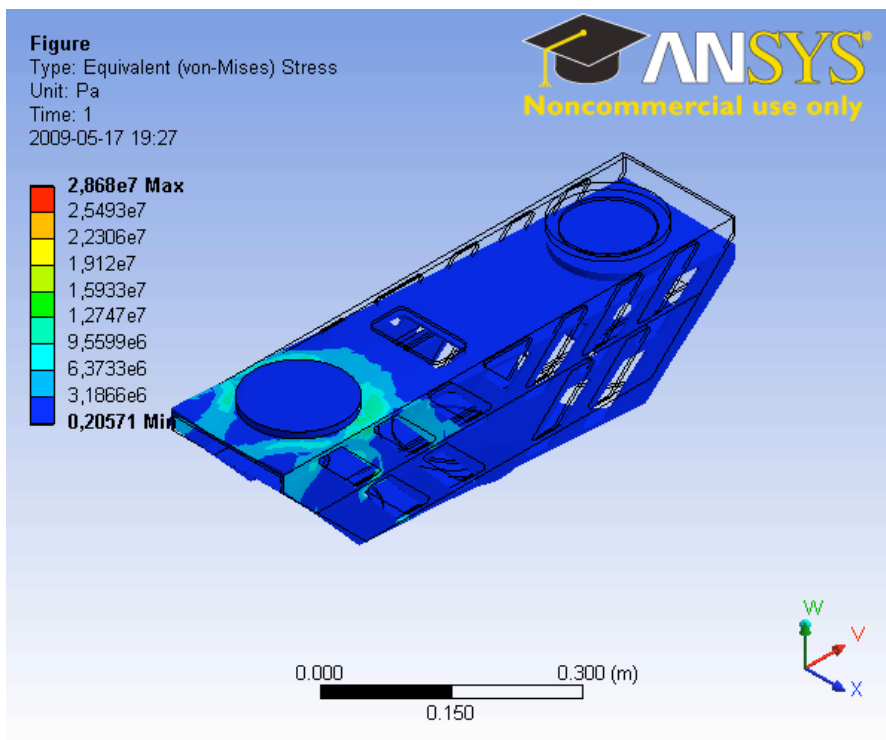
Figur 26: Förskjutningar i den artificiellt optimerade balken



Figur 27: Förskjutningar i den manuellt utformade balken



Figur 28: Spänningar i den artificiellt utformade balken



Figur 29: Spänningar i den manuellt utformade balken

7.4 Diskussion

Resultaten visar på att de genetiska algoritmerna lyckades konstruera en balk som hade lägre massa, mindre förvriddning av balken och lägre maximal spänning jämfört den med hjälp av ANSYS skapade balken. Jämfört med den mänskligt utformade balken är den mer effektivt utformad med en viktminskning på 23 %. Viktminskningen jämfört originalbalken är hela 36 %.

Detta visar på det utvecklade programmets styrka när det gäller att hitta lösningar på komplexa problem som tidigare krävt stor konstruktionsvana. Det visar också på svagheterna i dagens optimeringsprogram som ofta endast kan ge en vag bild av hur konstruktionen hade kunnat förbättras genom att visa de delar där material kan tas bort.

Med det utvecklade programmet kan användaren genom att ange randvillkor, krav och optimeringsmål snabbt få en bild av lämpliga strukturer som sedan kan överföras till en färdig lösning.

Optimering eller konstruktion med hjälp av genetiska algoritmer har alltså potential att skapa fullvärdiga och kreativa lösningar utan mänsklig inverkan. De brister som kan pekas på är vid finoptimeringen eftersom GA bygger på en rent stokastisk sökning. Det är möjligt att i kombination med en noggrannare optimeringsmetod vid slutet av optimeringen hade kunnat finna ännu bättre resultat.

Speglingsmutationen som infördes visade sig vara fördelaktig. Detta kan förklaras av att en struktur som fungerar bra i en del av balken även bör kunna vara fördelaktig i andra. Denna sorts utseende återfinns även i naturen där många djur och växter är symmetriska kring ett eller flera plan.

Mutationen som ersatte alla individer i populationen med mutationer av den bästa individen visade sig vara värdefull då lösningens utveckling hade stannat av. Genom att ersätta hela generationen med mutationer av den bästa individen kunde nya varianter hittas snabbare.

8. Konstruktion av voronoistruktur till bord

8.1 Inledning

De tidigare avsnitten har behandlat konceptgenerering och optimering av strukturer. I denna sista del knyts de samman för att skapa ett program som ger en potentiell kund möjlighet att utforma en produkt enligt eget tycke, utan att behöva ha kunskap om varken hållfasthetsberäkningar eller optimeringstekniker.

Normalt sett finns det få möjligheter för en kund att själv kunna anpassa en produkt utöver att välja bland ett fåtal fördefinierade storlekar och färger. Anledningen till detta har varit att det är kostsamt att specialanpassa varje enskild vara istället för att kunna massproducera den. Det är dyrt att tillverka nya verktyg som gjutformar eller pressverktyg. Dessutom krävs experter som kontrollerar att varan går att tillverka och fungerar i praktiken, samt att det inte går åt för mycket material. Ofta måste framställandet av individuella produkter ske genom ett längre samarbete mellan kund och tillverkare.

Men nya framsteg inom datorstyrd tillverkning (CNC) innebär numera att tio föremål med variationer i designen kostar lika mycket tid och pengar att tillverka som tio identiska föremål. (Hopkinson, Hague & Dickens, 2006)

Problemet som kravstår är då hur den mänskliga expertisen inom tillverkning, hållfasthet och kostnadsoptimering kan automatiseras.

Som visats i tidigare avsnitt kan en konstruktör med hjälp av det utvecklade programmet framgångsrikt angripa just sådana problem. Hittills har optimeringen skett utan inblandning av en mänsklig användare, men för att kunna anpassa produkten efter en kunds tycke och smak måste en sådan länk även skapas.

För att utnyttja det utvecklade programmets möjlighet att optimera komplexa strukturer och möjligheterna med CNC-tillverkning så inriktades produkten mot strukturer hämtade från naturen.

Strukturerna i naturen har framkommit under miljarder år som resultat av evolutionen. De har anpassats efter sin miljö och optimerats för att ge så stor chans till fortplantning som möjligt och därigenom blivit effektiva.

Det finns en lång tradition av att hämta inspiration och idéer från naturen inom arkitektur och konst. Inom produktutveckling har det varit svårt att utnyttja resultat från naturen för att skapa nya produkter på grund av att organiska och invecklade former inte är särskilt väl anpassade för massproduktion.

Strukturerna är även svåra att anpassa till nya tillämpningar eftersom anledningen till varför de fungerar som de gör ofta inte är helt kartlagda (Stach, 2004).

Dessa begränsningar är dock på väg att övervinnas i takt med att CNC-tillverkning blir allt mer mångsidig och med hjälp av GA som är väl lämpat för just sådana problem där lösningens utseende är helt okänt och många variabler spelar in enligt avsnitt 4.4.

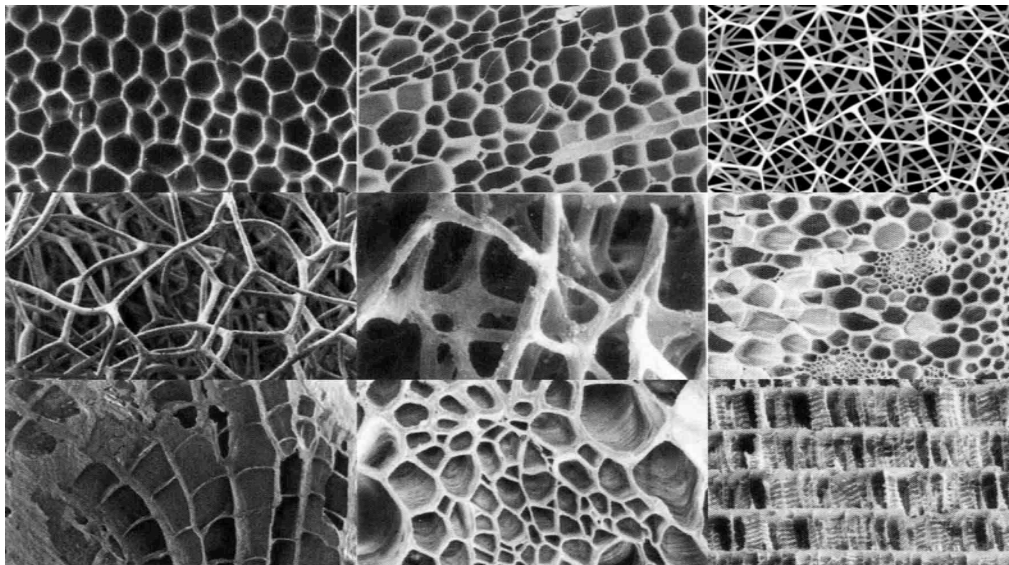
Arbetet genomfördes tillsammans med Andreas Hopf som är lektor vid institutionen för industridesign, LTH och som tidigare har forskat kring ytor och strukturer från naturen och matematiken. Han har länge önskat tillämpa dessa ytor i produkter, men har på grund av de svårigheter som tidigare beskrevs inte kunnat göra några större projekt.

8.2 Val av produkt

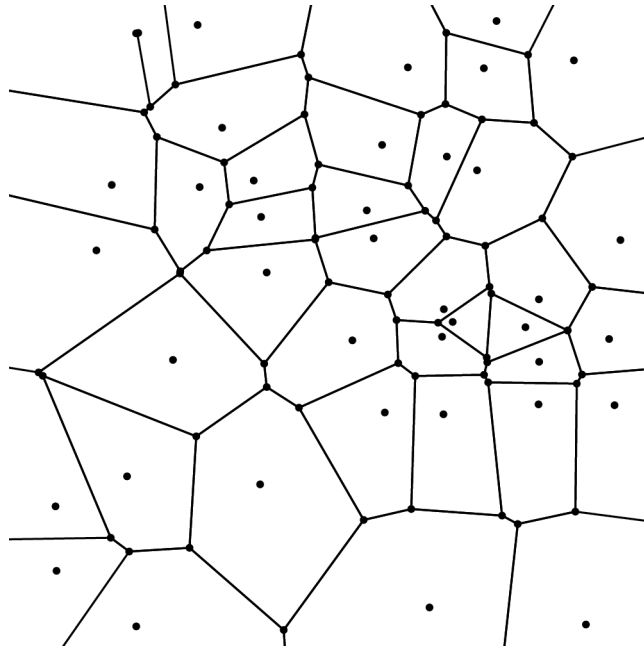
En av de vanligaste egenskaperna för strukturer i naturen är att de är starka i relation till sin vikt (Beukers & van Hinte, 2005), se till exempel till bikupor, växtstjälkar och skelettben. Detta är naturligtvis även en eftertraktad egenskap hos produkter i allmänhet. Se till exempel Figur 30.

Dessa strukturer kan ofta beskrivas av Voronoidiagram (Okabe et al., 1992) som skapar ett mönster enligt Figur 31. Diagrammet skapas genom att för varje voronoipunkt s rita ut det områden där alla punkter är närmast s och ingen annan voronoipunkt (Dirichlet, 1850).

En produkt som har stort behov av styvhet och låg vikt är bordet, och därför valdes alltså ett bord med voronoistruktur som produkt.



Figur 30: Naturligt lätta och starka strukturer från naturen som kan beskrivas av voronoidiagram



Figur 31: Voronoidiagram

8.3 Tillverkningsmetod

För att kunna modellera bordets struktur för femanalys så var det viktigt att ha en ungefärlig bild av hur det skulle komma att tillverkas och monteras. De metoder som övervägdes var:

1. Laserskärning av plåtremсор och robotsvetsning av dem i noderna
2. Laserskärning av plåtremсор som sedan bockas till enskilda celler som sedan fogas till varandra genom skruv-, nit-, eller svetsförband
3. Sickning av en plåt längs med cellväggarna

Ett liknande bord hade tidigare skapats genom metod 2 av Andreas Hopf. Metoden möjliggör att cellerna inte måste vara färdigmonterade vid transport, utan kan sättas samman av kunden. Det gör även att cellerna kan tillverkas i olika färger och material och kan bytas ut och flyttas runt. Detta är inte möjligt med de andra teknikerna som ger ett färdigt och fast monterat bord direkt vid tillverkningen. Eftersom transporterbarhet, monterbarhet och möjlighet till anpassning av bordets utseende bedömdes som viktigt valdes denna metod. Samtidigt ligger metoden ganska nära metod 1 vilket gör att den kan användas som ett alternativ utan att några ändringar behöver göras.

Valet av material var aluminium med materialdata enligt Tabell 5.

Tabell 5: Materialdata

Elasticitetsmodul	68,95 GPa
Densitet	2710 kg/m ³
Poissions ratio	0,3
Sträckgräns	280 MPa

8.4 Problemformulering

Skapa ett program där en kund kan bestämma produktens huvudsakliga utseende med hjälp av GA. Sedan skall produktens struktur optimeras för att klara vissa produktions- och hållfasthetskrav och minimera produktens kostnad.

Största nedböjning vid belastning av sin egen vikt är satt till 3 mm för att ge en stabil bordsyta och undvika att bordet deformeras synbart. Tillverkningsmetoden tillför ytterligare krav på strukturen i form av kortaste cellvägg och minsta vinkel mellan två väggar. En förfrågning hos ett tillverkningsföretag (Ferrum Lasercut GmbH, Berlin) gav 30 mm som kortaste cellvägg och 33° som minsta vinkel, se Figur 33. Dessa begränsningar var enligt tillverkningsföretaget snarlika för alla företag.

Optimeringsmålet är att minimera produktens kostnad, vilket främst bedöms påverkas av mängden använt material, det vill säga produktens massa. Valet av tillverkningsmetod gör att monteringskostnaden utesluts.

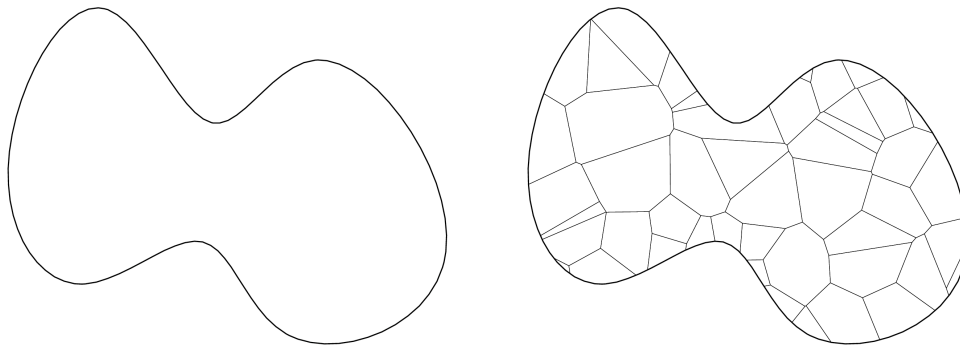
8.5 Implementering

8.5.1 Kundenspecificerade parametrar

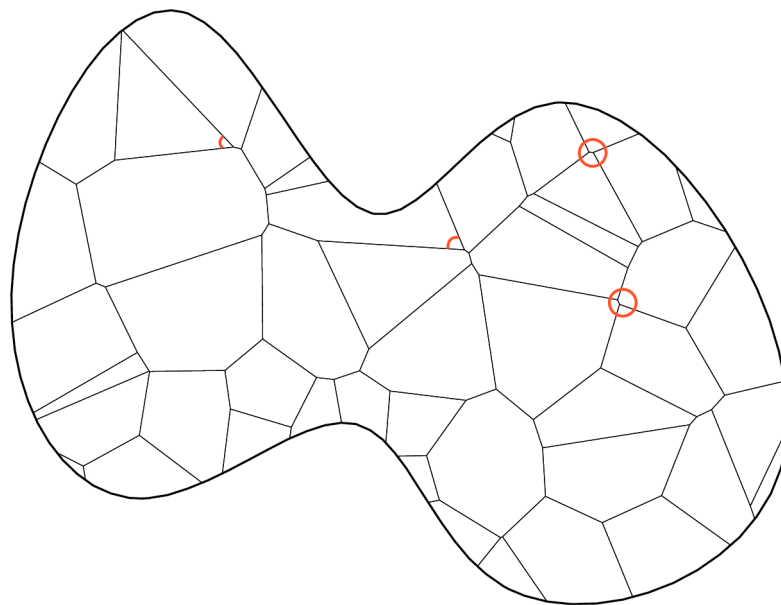
För att ge kunden möjlighet att anpassa bordets utseende, utan att behöva göra några detaljerade inställningar beslutades det att de viktigaste parametrarna för kunden var att kunna bestämma bordets höjd, yttre kontur, placering av benen och bordsskivans tjocklek. Kunden hade även möjlighet att välja gränser för vissa parametrar som påverkar voronoistrukturens utseende.

8.5.2 Skapandet av bordsstrukturen

Efter att kunden eller en designer hade definierat konturen till bordet med hjälp av ett enkelt hjälpprogram skapas en voronoistruktur efter konturen. Se Figur 32.



Figur 32: Den av kunden definierade konturen till bordet och den skapade voronoistrukturen

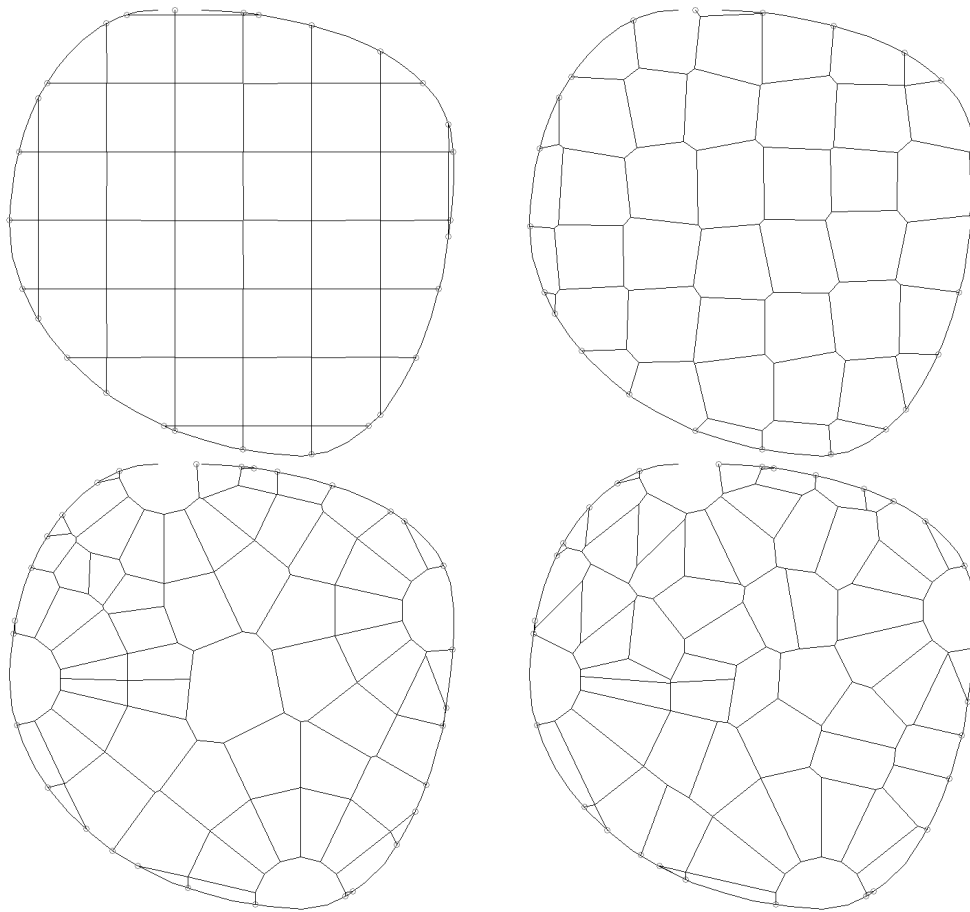


Figur 33: Problemområden vid tillverkning i form av korta väggar och små vinklar

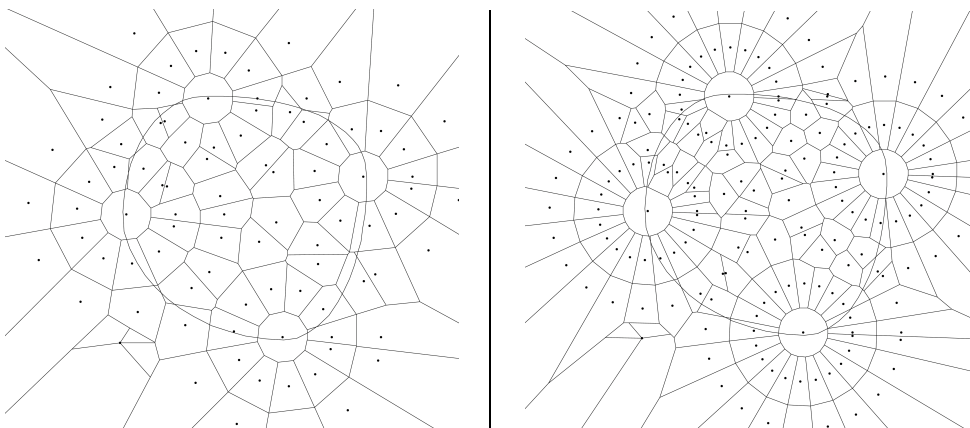
Voronoidiagrammet skapas genom en algoritm vilken först lägger ett rutnät av punkter över den kontur som kunden definierat. Den skapade även en koncentration av punkter kring varje ben. För varje ben skapades en punkt på dess position.

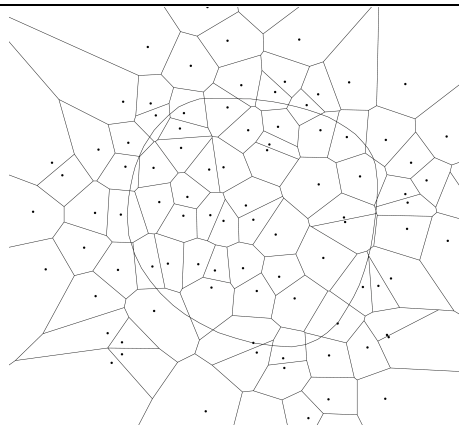
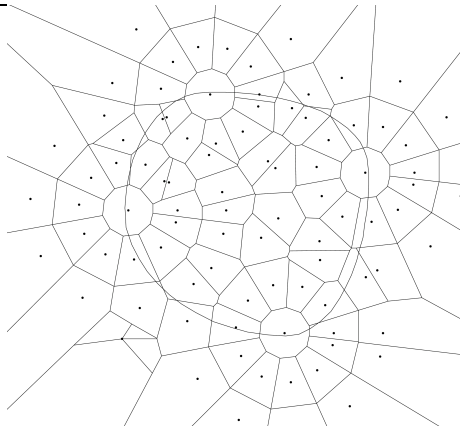
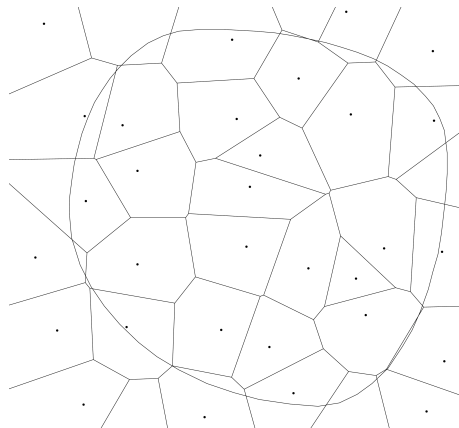
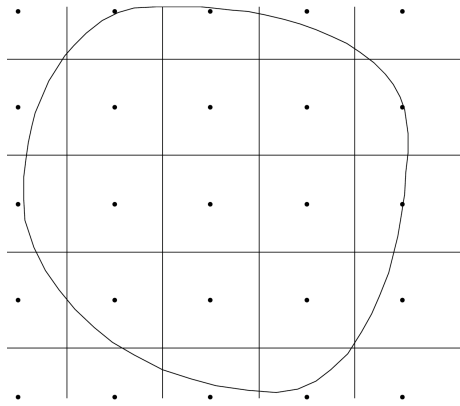
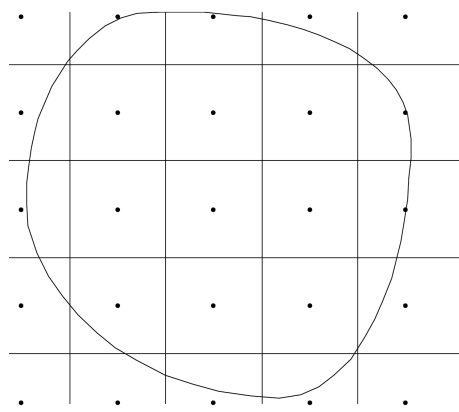
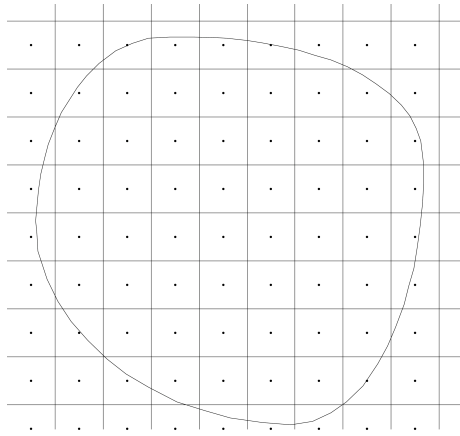
De variabler som kunde varieras av algoritmen var radien av utspridda punkter kring benen, rutnätets täthet och hur slumpmässigt punkterna placerades i rutnätet respektive kring benen, se Figur 34 och Figur 35. Totalt kunde alltså sex olika variabler ändras för att påverka bordets struktur vilka alltså utgjorde bordets *genom*.

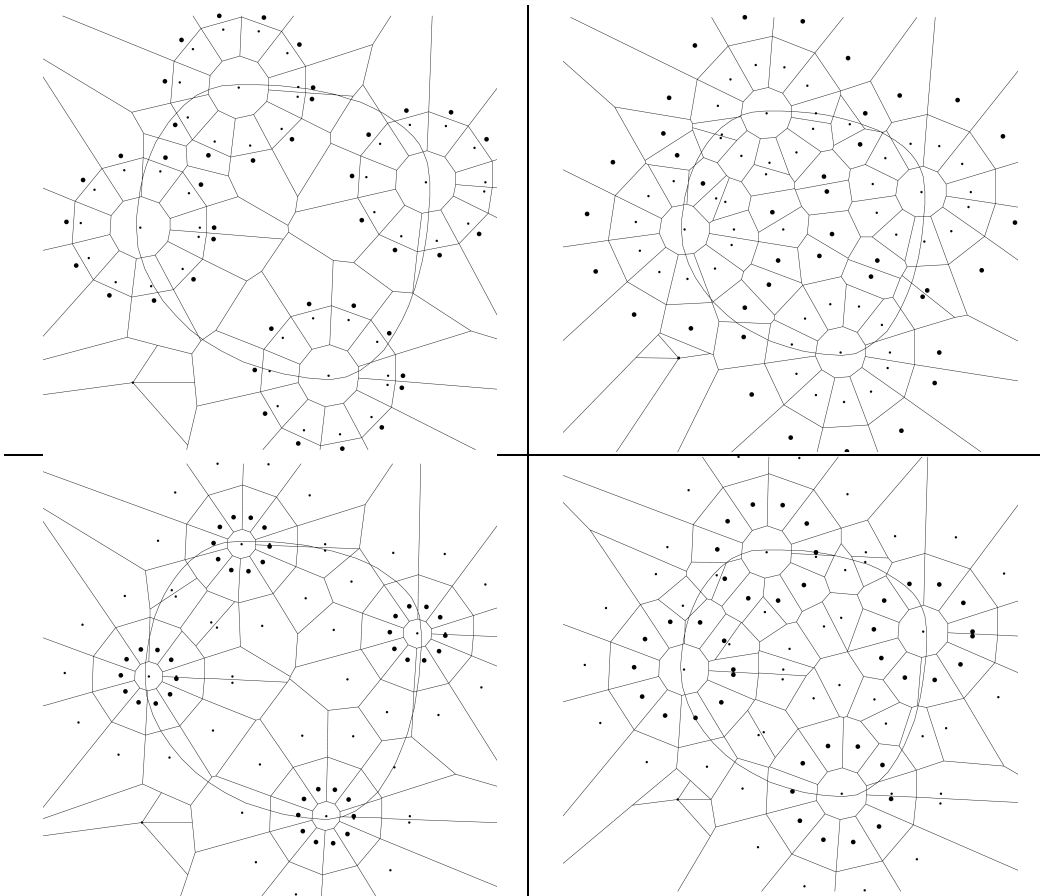
Metoder för att skapa voronoidiagram utifrån angivna punkter fanns redan i *MATLAB*.



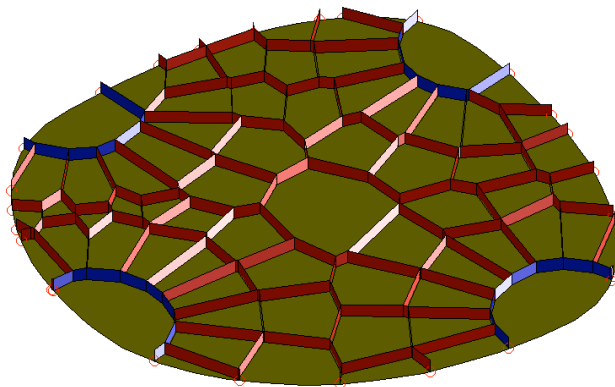
Figur 34: Demonstration av hur voronoistrukturen skapades. Först skapades en matris av punkter som bildade ett rutnät, sedan tillfördes slumpmässiga värden till punkterna. Därefter skapades punkter kring benen och till sist lades punktmatrixen och punkterna kring benen ihop.



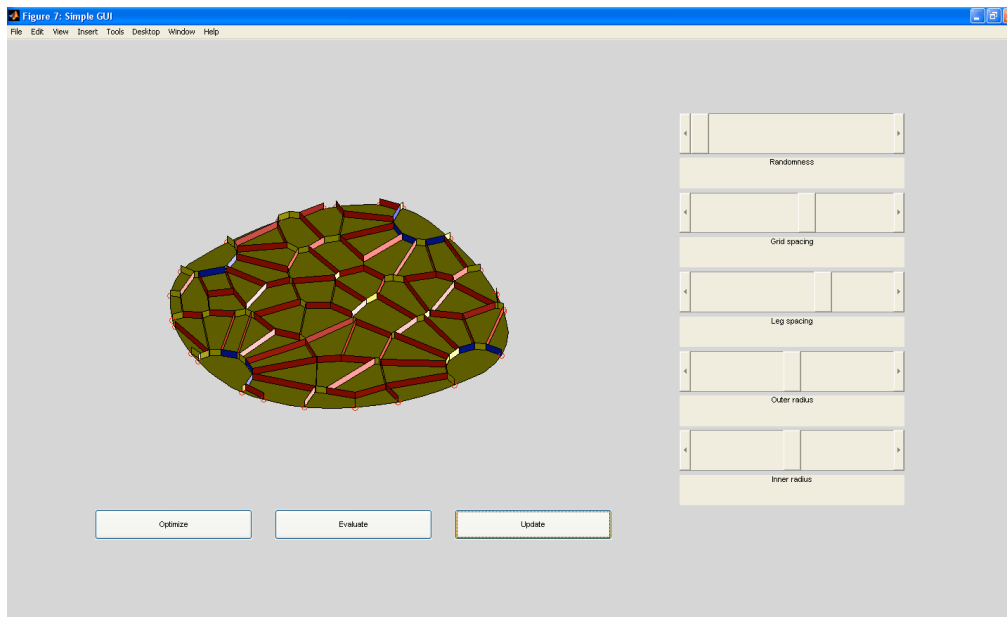




Figur 35: Illustration av de sex variablerna för bordets utseende. De illustrerade variablerna är i ordning: antal cirkelsegment, rutnätets indelning, rutnätets slumpmässighet, cirkelsegmentens slumpmässighet, cirkelsegmentens yttre radie och inre radie



Figur 36: Bordstrukturen i 3D. Blå cellväggar är tänkta som början på ben



Figur 37: Program för att ställa in olika utgångsparametrar. I gult visas cellväggar som har för korta längder.

8.5.3 Variabler för optimeringen

De variabler som bedömdes vara intressanta för algoritmen att kunna mutera var de som beskrevs i avsnitt 8.5.2. Även tjockleken på plåten och cellernas höjd övervägdes kunna vara varierbara men celler med olika höjd ansågs olämpligt ur en estetisk synvinkel. Antalet möjliga plåttjocklekar är inte så stort och beror på tillverkningsmetoden för bordet.

Totalt bestämdes alltså varje individs struktur av sex olika variabler som kunde varieras enligt Figur 35.

8.5.4 Representation och modellering av strukturen

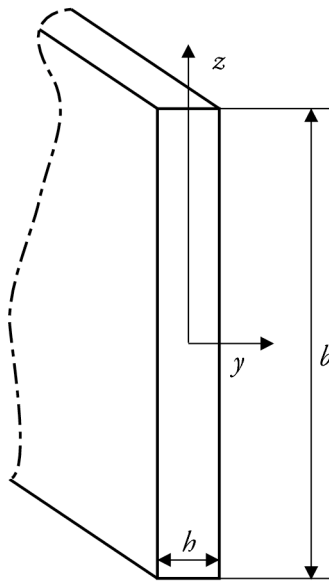
Om konstruktionen tillverkas som beskrivet i avsnitt 8.3 så är varje cellvägg fastsvetsad i en annan. Detta innebär att cellväggarna måste kunna ta upp moment. I CALFEM finns en inbyggd modell för denna sorts element, ramverkselement, vilken beslutades användas istället för de tidigare använda fackverkselementen. Ett ramverkselement kan förutom normalkrafter, som de tidigare använda fackverkselementen tog upp, även ta upp moment och skjuvkrafter. Varje cellvägg kunde alltså beskrivas som ett ramverkselement. Den enda förändringen i programmet med anledning av övergången till ramverkselement var att varje element nu representerades av sex frihetsgrader, tre för krafter i x -, y -, z -led, och tre för moment kring x -, y -, z -axlarna. I övrigt var representationen den samma som tidigare.

Noderna till de cellväggar där bordsbenen var tänkta att fästas sattes som fast inspända. Bordskivans vikt beräknades och lasten fördelades jämnt över alla bordets noder.

8.5.5 Utvärderingsfunktion

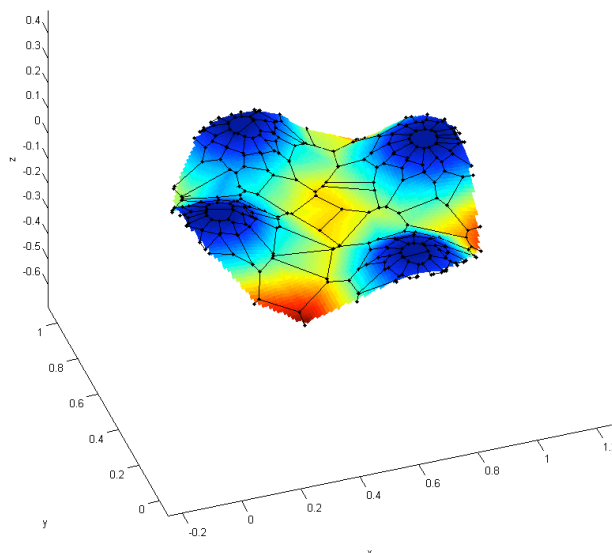
För att beräkna balkelementens förskjutningar behövs tröghetsmomentet I_y , tröghetsmomentet I_z , skjuvmodulen G , och vridstyvhets tvärsnittsfaktor K_v för varje element. För att simulera en plåtremsas tvärsnitt användes det tvärsnitt som visas i Figur 38. Enligt Sundström (1998) kan de nämnda storheterna beräknas med hjälp av:

$$I_y = \frac{hb^3}{12}, I_z = \frac{bh^3}{12}, G = \frac{E}{2(1+\nu)}, K_v = \frac{bh^3}{3}$$



Figur 38: Tvärsnitt av ett balkelement

För att bättre kunna visualisera deformationen av bordet interpolerades en yta efter bordets förskjutna noder och lades som en duk över diagrammet, se Figur 39.



Figur 39: Visualisering av femberäkningen. Deformationen av bordet är kraftigt överdriven

Tidigare har poängen satts efter konstruktionens vikt, alltså optimeringsmålet. Om konstruktionen inte har uppfyllt kraven för lösningen så har den fått poäng 0. Tidigare har detta fungerat bra eftersom det är sällan som konstruktionerna inte har klarat kraven.

Eftersom antalet krav på strukturen nu är fler än innan fick poängsättningen av individerna revideras. Nu får lösningen istället först poäng för hur nära den är att uppfylla kraven. På detta vis drivs utvecklingen framåt mot ett bord som klarar kraven.

Optimeringen sker i tre steg. Först för att lösningen skall klara av kravet på största nedböjning, sedan kravet på minsta längd och vinkel, och sist optimering av vikten. Ordningen är sådan eftersom strukturer som först optimeras med hänsyn till kravet på tillverkbarhet ofta genererar lösningar som har långa cellväggar men är instabila.

8.5.6 Genetiska operatorer

8.5.6.1 Mutation

Endast en enkel mutering av värdet för de olika variablerna beskrivna i 8.5.3 skedde. Detta genom att helt enkelt lägga till ett slumpmässigt värde till slumpmässigt utvalda parametrar. Ju senare generation desto mindre ändringar gjordes.

8.5.6.2 Korsning

Även här användes en grundläggande korsning som utbytte slumpmässigt utvalda parametrar med sina motparter i två individer.

8.5.7 Sökparametrar

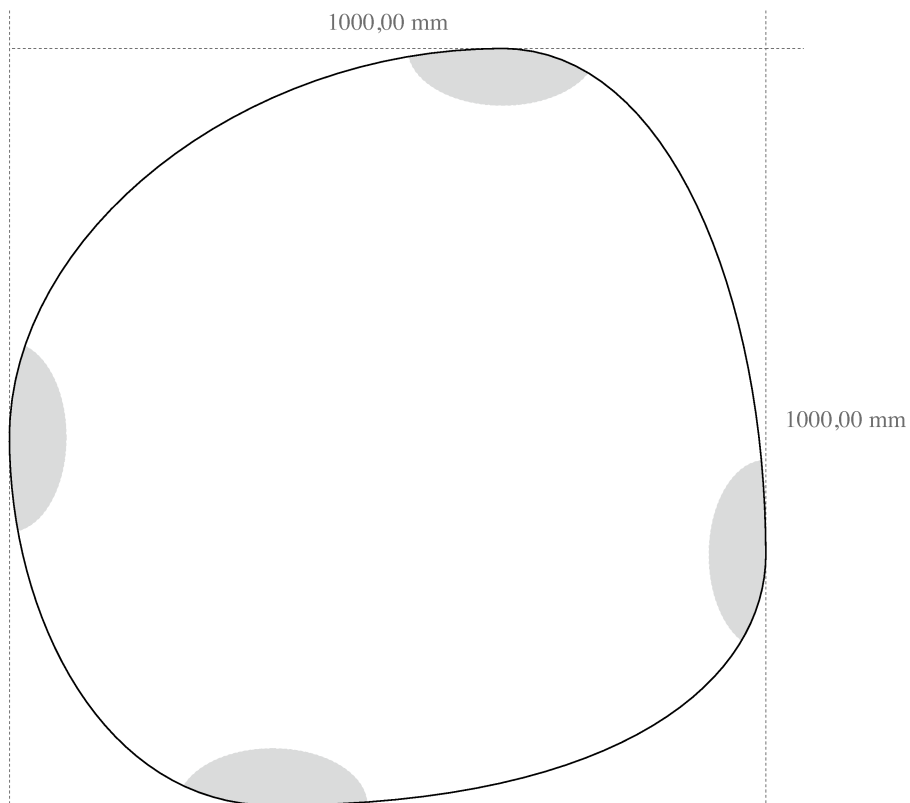
Här användes 50 individer och 500 generationer. Dock tog optimeringen denna gång betydligt längre tid på grund av att varje individ måste skapas på nytt varje gång Voronoidiagrammets ändras eftersom en ändring i en voronoipunkt även påverkar alla omkringliggande cellväggar.

8.6 Resultat

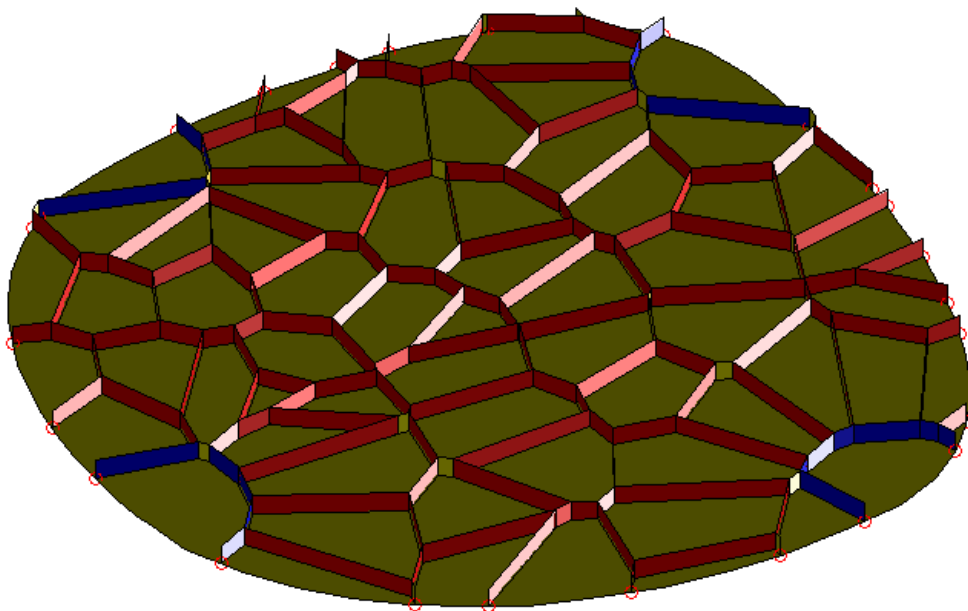
En kontur utformades enligt Figur 40. En utgångsstruktur för bordet visas i Figur 41. I Figur 42 visas evolutionen av bordet. I Figur 43 och Figur 44 visas resultatet av optimeringen. I Figur 45 till Figur 47 visas den slutgiltiga designen efter att vissa estetiska modifieringar gjorts. De förändringar som gjordes var att låta alla celler i ytterkanten utgöra ben, och att göra övergången mellan benen och de andra cellerna mjukare. I Tabell 6 visas en jämförelse av bordet före och efter optimeringen. Cellernas höjd är 50 mm och cellväggarnas tjocklek är 1 mm.

Tabell 6: Jämförelse av bordets egenskaper före och efter optimering. a_{\max} är största nedböjning i m, w är bordets vikt, l_{\min} är kortaste cellvägg i m och α_{\min} är minsta vinkel mellan två cellväggar

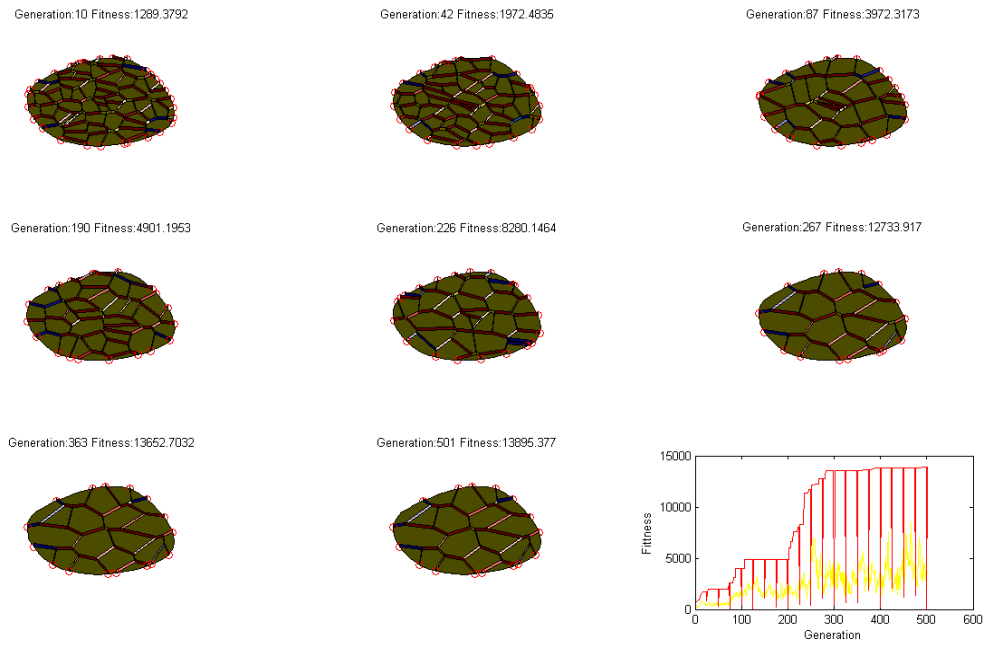
	Före optimeringen	Efter optimering
a_{\max}	0,00049386	0,0021
w	5,3681	2,6779
l_{\min}	0,0018	0,0347
α_{\min}	32,2923	40,88



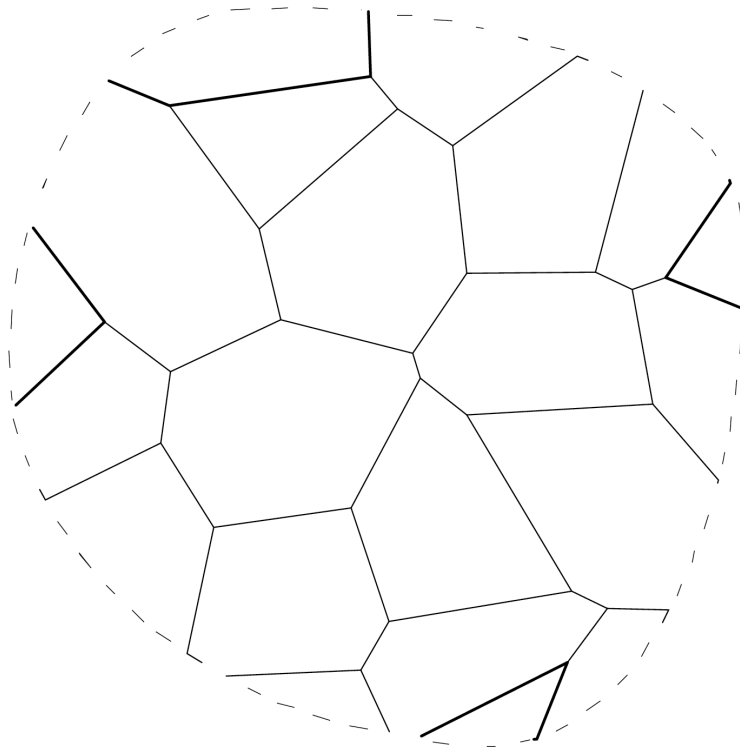
Figur 40: Den valda konturen



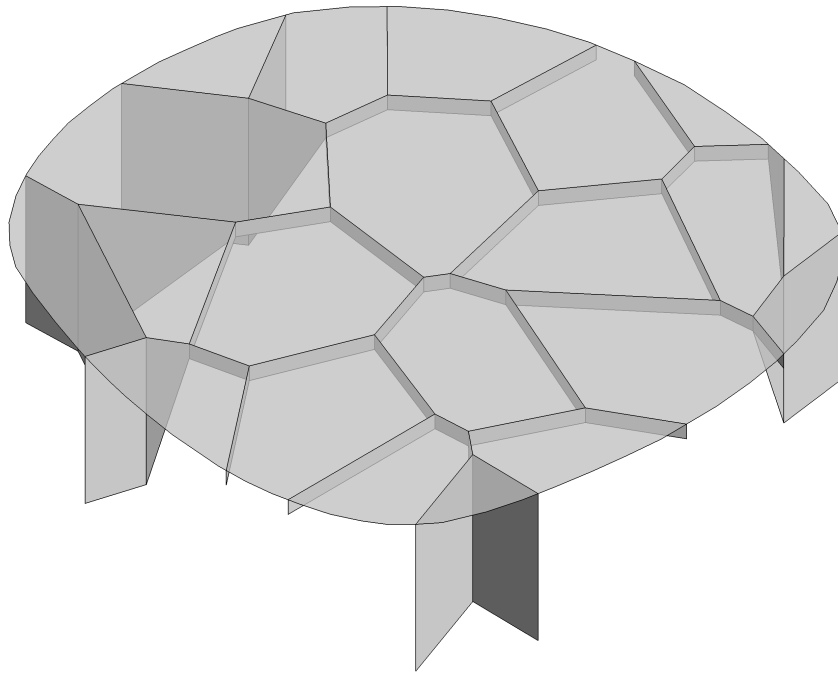
Figur 41: Utgångsstrukturen



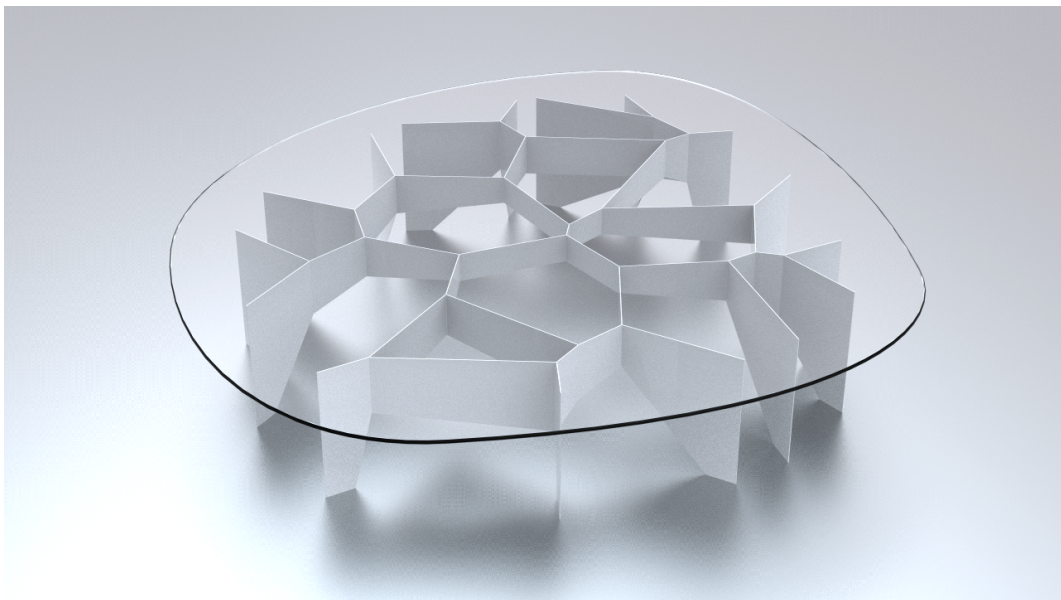
Figur 42: Evolution av strukturen under 500 generationer



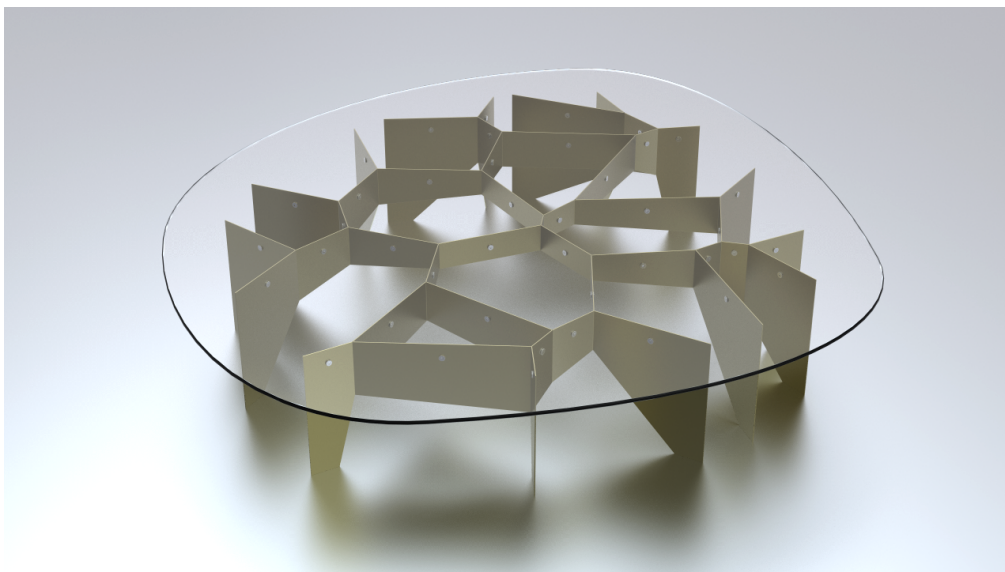
Figur 43: Toppvy av konstruktionen. I feta linjer visas de celler som har stöd underifrån av ben



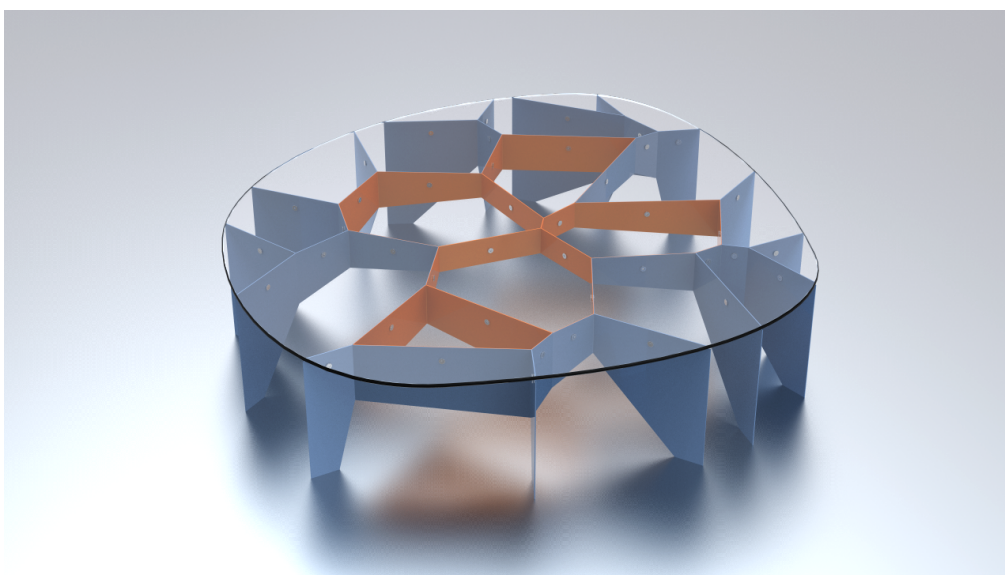
Figur 44: Bordsstrukturen som algoritmen genererade



Figur 45: Den slutgiltiga konstruktionen efter att några estetiska modifikationer gjorts



Figur 46: Ett utförande i mässing



Figur 47: Ett utförande med olikfärgade celler

8.7 Diskussion

Det slutgiltiga bordet som uppkom uppfyllde alla krav som var ställda på det och hade en vikt som minskat med mer än hälften från utgångsstrukturen.

Det som från början uppskattades som det svåraste att uppnå var undvika att nedböjningen blev för stor. Detta krav uppfylldes redan i de första generationerna, förmodligen på grund av att voronoidiagram ofta skapar starka strukturer (Okabe et al., 1992). Istället var det att skapa en struktur med tillräckligt långa cellväggar som var det

mest tidskrävande. Det är först i generation 250 som det kravet uppfyllts. Detta visar på hur svårt det tidigare varit att konstruera strukturer från naturen som gått att tillverka.

Med hjälp av GA och modern tillverkningsteknik kan alltså helt nya produkter skapas, som tidigare varit enormt tidskrävande eller rent utav omöjliga att utforma. Nu kan den enorma källan av optimerade strukturer och ytor från naturen äntligen börja utnyttjas till produktutveckling och industridesign. Detta bord kan mycket väl vara början till en renässans inom produktdesign där teknik och natur möts.

9. Slutsatser och rekommendationer

De resultat som framkommit under arbetets gång visar på styrkan bakom att använda en evolutionär optimeringsmetod för att skapa nya produkter.

I kapitel 6 visades hur evolutionära optimeringsmetoder kan underlätta för konstruktören vid konceptgenereringen. Genom att endast ange randvillkor och krav på en struktur kan ett antal olika lösningar på samma problem genereras. Många av vilka uppvisade stora likheter med mänskligt utformade broar. Variationen i lösningarna är ett resultat av optimeringsmetodens stokastiska natur. Används en deterministisk metod erhålls alltid samma resultat vilket inte är särskilt användbart vid konceptgenerering där det är viktigt att kunna skapa många olika alternativa lösningar.

I kapitel 7 angreps ett vanligt problem inom produktutveckling, viktoptimering av en balkstruktur. En jämförelse gjordes med en idag vanlig teknik– att stegvis förändra strukturen manuellt och utvärdera resultaten med hjälp av ett strukturanalysprogram, i detta fall ANSYS, för att se var ytterligare material kan tas bort.

Resultatet här är en viktminskning på 23 % jämfört den mänskligt utformade balken med lägre förvriddning av balken och lägre maximala spänning i strukturen.

I kapitel 8 beskrivs hur en komplex struktur från naturen anpassats till helt nya tillämpningar. Detta har inte tidigare varit möjligt inom produktdesign då arbetet med att tillämpa naturliga strukturer på ett kostnadseffektivt, tillverkbart och hållfast sätt varit en alltför krävande uppgift för en människa.

Med det utvecklade programmet kan nu vilken person som helst göra denna tidigare omöjliga bedrift. Allt som behövs är en kontur till bordet och ett intervall för parametrarna som används vid skapandet av bordets struktur. Sedan optimeras automatiskt ett bord som uppfyller alla de ställda kraven, och som kräver så lite material som möjligt. Bordet är sedan färdigt att tillverkas med datorstyrda metoder, vilket möjliggör att varje kund kan få ett individuellt anpassat bord.

Inom detta område där natur, matematik, modern tillverkningsteknik och produktdesign möts är möjligheterna till fortsatt utveckling stora. Det finns en enorm mängd intressanta och effektiva lösningar från naturen som nu kan börja utnyttjas i full skala genom denna sorts evolutionär metoder och mångsidiga tillverkning.

Ett intressant område för framtida forskning är att undersöka hur effektiva lösningar från naturen kan utnyttjas inom produktdesign med hjälp av evolutionära optimeringsmetoder och modern tillverkningsteknik.

Referenser

ANSYS, I., [Online], Tillgänglig: <http://www.ansys.com/>.

Austrell, P., Dahlblom, O., Lindemann, J., Olsson, A., Olsson, K., Persson, K., Petersson, H., Ristinmaa, M., S, berg, G. & Wernberg, P. (2004) *CalfeM a finite element toolbox*, Lund.

Beukers, A. & van Hinte, E. (2005) *Lightness: the inevitable renaissance of minimum energy structures*, 010 Publishers.

Black, H.S. (1977) 'Inventing the negative feedback amplifier', *IEEE Spectrum*, vol. 14, no. 12, pp. 54-60.

Brown, D.C. (1987-1997) 'Artificial Intelligence for Engineering Design, Analysis and Manufacturing', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*.

Cahon, S., Melab, N. & Talbi, E.-G. (2004) 'ParadisEO: A Framework for the Reusable Design of Parallel and Distributed Metaheuristics', *Journal of Heuristics*, vol. 10, no. 3, May, pp. 357-380.

Coello, C. & Christiansen, A. (2000) 'Multiobjective optimization of trusses using genetic algorithms', *Computers and Structures*, vol. 75, no. 6, pp. 647-660.

COMSOL *COMSOL - Multiphysics Modeling*, [Online], Tillgänglig: <http://www.comsol.com/>.

Cuthill, .E. & McKee, .J. (1969) 'Reducing the bandwidth of sparse symmetric matrices', Proceedings of the 1969 24th national conference, New York, NY, USA, 157-172.

Dirichlet, G.L. (1850) 'Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen', *J. reine angew. Math*, vol. 40, pp. 209-27.

Eby, D., Averill, R., Punch, W.F. & Goodman, E.D. (1999) 'Optimal design of flywheels using an injection island genetic algorithm', *AI EDAM*, vol. 13, no. 05, pp. 327-340.

Finnman, K.O. (2009) *In progress*, Lund.

Giger, M. & Ermanni, P. (2006) 'Evolutionary truss topology optimization using a graph-based parameterization concept', *Structural and Multidisciplinary Optimization*, vol. 32, no. 4, pp. 313-326.

Holland, J.H. (1975) *Adaptation in natural and artificial systems*, MA: MIT Press Cambridge.

- Hopkinson, N., Hague, R. & Dickens, P. (2006) *Rapid manufacturing: an industrial revolution for the digital age*, John Wiley & Sons.
- Houck, C.R., Joines, J. & Kay, M. (1995) 'A Genetic Algorithm for Function Optimization: A Matlab Implementation', *NCSU-IE TR*, vol. 95, no. 09.
- Janberg, N. (1998-2009) *Structurae*, [Online], Tillgänglig: <http://en.structurae.de/> [2009].
- Jun, L., Zhenping, F., Jianzhong, C. & Zuda, S. (1996) 'Aerodynamic optimum design of transonic turbine cascades using Genetic Algorithms', *Journal of Thermal Science*, vol. 6, November.
- Kawamura, H., Ohmori, H. & Kito, N. (2002) 'Truss topology optimization by a modified genetic algorithm', *Structural and Multidisciplinary Optimization*, vol. 23, no. 6, pp. 467-473.
- Kelesoglu, O. (2007) 'Fuzzy multiobjective optimization of truss-structures using genetic algorithm', *Advances in Engineering Software*, vol. 38, no. 10, pp. 717-721.
- Kirk, B., Peterson, J.W., Stogner, R.H. & Carey, G.F. (2006) 'libMesh: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations', *Engineering with Computers*, vol. 22, no. 3-4, pp. 237-254.
- Koza, J.R., Keane, M.A., Streeter, M.J., Adams, T.P. & Jones, L.E.E.W. (2005) 'Invention and creativity in automated design by means of genetic programming', *AI EDAM*, vol. 18, no. 03, pp. 245-269.
- Lohn, J.D., Linden, D.S., Hornby, G.S., Kraus, W.F., Rodriguez-Arroyo, A. & Seufert, S.E. (2003) 'Evolutionary design of an X-band antenna for NASA's space technology 5 mission', *Evolvable Hardware, 2003. Proceedings. NASA/DoD Conference on*, 155-163.
- MathWorks, T. *The MathWorks - MATLAB and Simulink for Technical Computing*, [Online], Tillgänglig: <http://www.mathworks.com/>.
- Okabe, A., Boots, B., Sugihara, K. & Chiu, S.N. (1992) *Spatial tessellations: concepts and applications of Voronoi diagrams*, New York: Wiley New York.
- Parmee, I.C. & Bonham, C.R. (2000) 'Towards the support of innovative conceptual design through interactive designer/evolutionary computing strategies', *AI EDAM*, vol. 14, no. 01, pp. 3-16.
- Pegram, G.H. (1885) *Truss for bridges*.
- Poli, R., Langdon, W.B. & McPhee, N.F. (2008) *A field guide to genetic programming*, Lulu Press.

Russell, S. & Norvig, P. (1995) *Artificial Intelligence: A Modern Approach*, Englewood Cliffs, New Jersey: Prentice Hall.

Stach, E. (2004) 'Form-optimizing processes in biological structures-self-generating structures in nature based on pneumatics', IASS International Symposium on Shell and Spatial Structures from Models to Realization. Montpellier: IASS.

Stolpe, M. & Svanberg, K. (2003) *Structural and Multidisciplinary Optimization*, vol. 25, pp. 62-64.

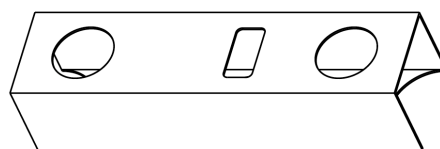
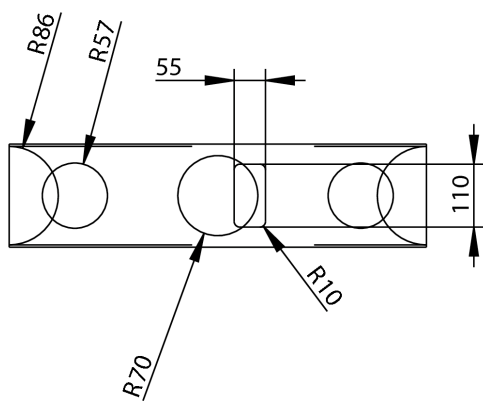
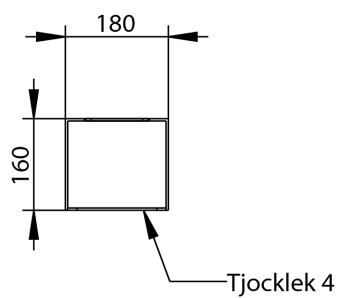
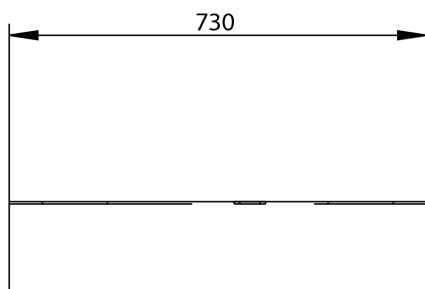
Sundström, B.a.K.t.h. (1998) *Handbok och formelsamling i hållfasthetslära*, Stockholm: Institutionen för hållfasthetslära, Tekniska högsk.

V. Toğan, A.T.D. (2006) 'Optimization of 3d trusses with adaptive approach in genetic algorithms', *Engineering Structures*, vol. 28, no. 7, pp. 1019-10

27.

Bilaga 1

Alla mått i mm

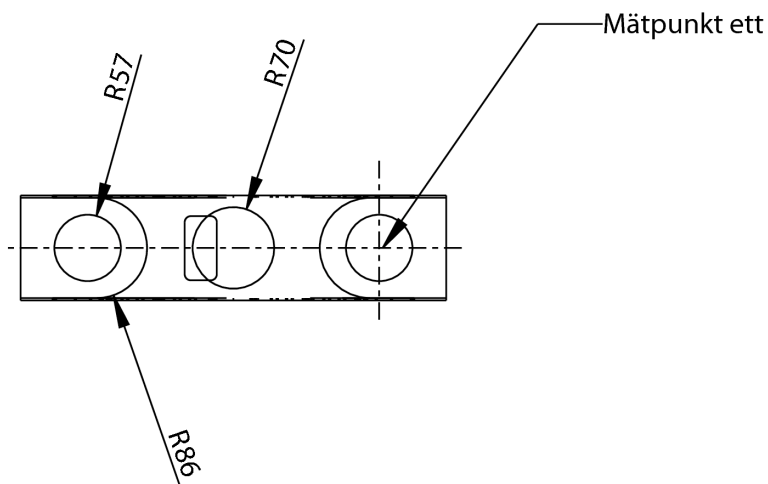
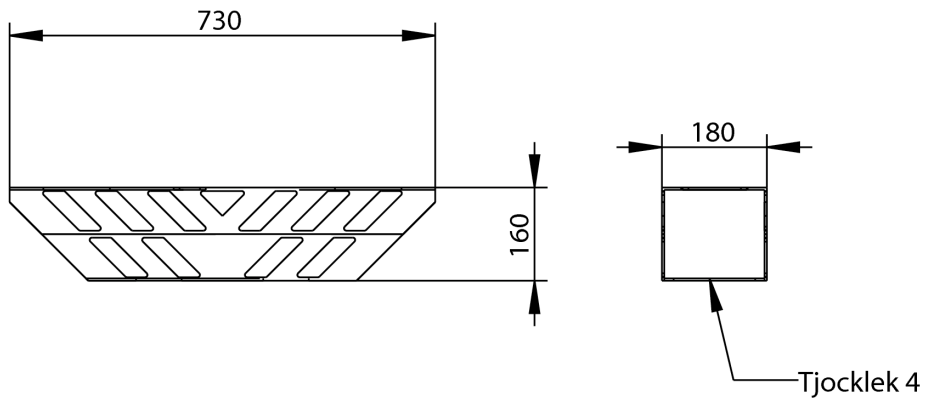


1:10

Utgångsstrukturs utseende

Bilaga 2

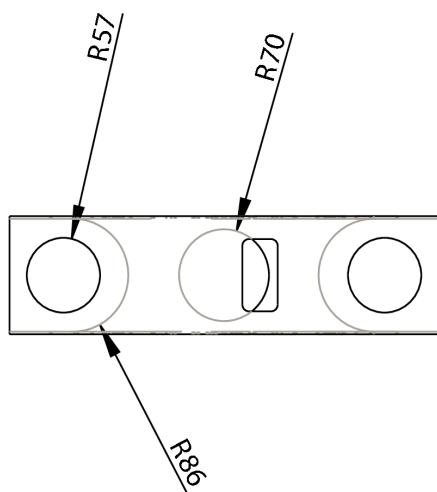
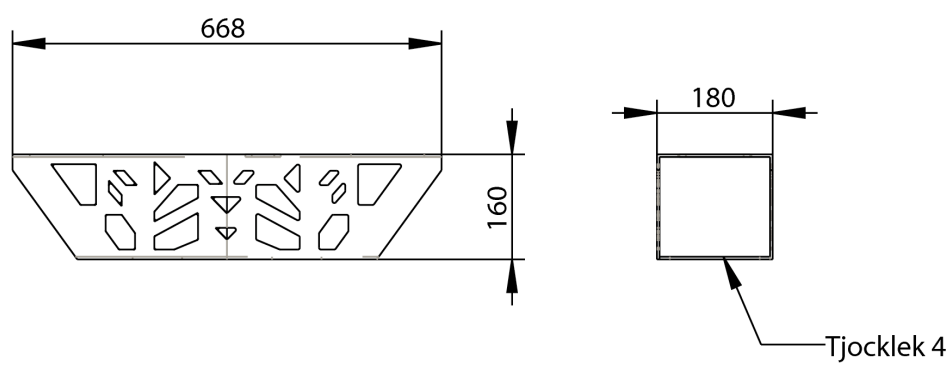
Alla mått i mm



Ritning av den mänskligt utformade balken med mätpunkterna för förskjutningar vid böjning och vridning

Bilaga 3a 3

Alla mått i mm



Ritning av den artificiellt utformade balken