PARVEILUALGORITMI KANTAVIEN RAKENTEIDEN OPTIMOINNISSA

Jussi Jalkanen

Rakenteiden Mekaniikka, Vol. 39
No. 2, 2006, ss. 23-35

TIIVISTELMÄ

Artikkelissa esitellään kantavien rakenteiden optimoinnin tehtävätyypit ja käydään läpi joitakin optimoinnin kannalta merkityksellisiä ongelmien piirteitä. Lukuisista mahdollisista eri ratkaisumenetelmistä käsiteltäväksi on valittu heuristinen parveilualgoritmi, joka periaatteessa sopii kaikentyyppisten kantavien rakenteiden optimoinnin tehtävien ratkaisemiseen. Parveilualgoritmin käyttöä havainnollistetaan kahdella esimerkkitehtävällä, jotka käsittelevät tasoristikon sauvojen poikkipinta-alojen valintaa ja teräskotelon muodon sekä seinämän paksuuksien optimointia. Kotelon optimointiongelmassa käytetään rakenteen analysoinnissa kaupallista FEM-ohjelmaa.

JOHDANTO

Tässä esityksessä tarkastellaan kantavien rakenteiden optimointiongelman ratkaisemista parveilualgoritmia (particle swarm optimization, PSO) käyttäen. Ideana on esitellä rakenteiden optimoinnin tarjoamia mahdollisuuksia erityyppisissä suunnitteluongelmissa. Optimointi on hyvin luonnollinen jatke rakenteiden tietokoneavusteiselle analysoinnille. Suunnittelijan ei kannata tyytyä muutaman vaihtoehdon keskinäiseen vertailuun FEM-laskelmien perusteella, vaan on paljon järkevämpää pistää tietokone hakemaan optimoinnin avulla uusia entistä parempia, mutta ennalta vaikeasti arvattavia ratkaisuja.

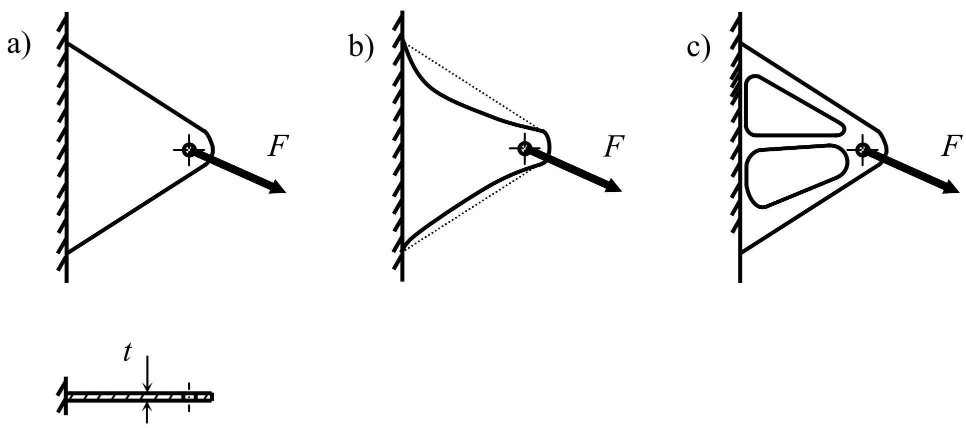
Parveilualgoritmi on perusidealtaan suhteellisen yksinkertainen ja melko tuore heuristinen optimointimenetelmä. Se sopii hyvin varsinkin diskreettejä suunnittelumuuttujia sisältävien laskennallisesti työläiden optimointiongelmien likimääräiseen ratkaisemiseen. Vaikka parveilualgoritmi ei pystykään takaamaan globaalin optimin löytymistä, päädytään usein hyviin ratkaisuihin, jotka ovat selvästi parempia kuin mitä parhaat siihen mennessä tiedetyt ovat olleet. Monissa käytännön sovelluksissa ei olla kiinnostuneita kaikkein parhaimmasta ratkaisusta, vaan sitä lähellä oleva kohtuullisella laskentatyöllä saatava ratkaisu on houkuttelevampi vaihtoehto.

Parveilualgoritmia käytettäessä rakenteen analysointi voidaan haluttaessa tehdä kaupallisella FEM-ohjelmistolla. Tällöin rakenne on mahdollista mallintaa varsin realistisesti käyttäen periaatteessa kaikkia ohjelman tarjoamia ominaisuuksia. Parveilualgoritmia käytettäessä riittää, että annetuilla suunnittelumuuttujien arvoilla pystytään tekemään rakenneanalyysi ja laskemaan kohdefunktion ja rajoitusehtofunktioiden arvot analyysin tulosten perusteella. Monen optimointialgoritmin tarvitsemia kohdefunktion ja rajoitusehtofunktioiden derivaattoja suunnittelumuuttujien suhteen ei tarvita. Yksi ajo ei saa kuitenkaan kestää kovin kauan, sillä analysointi tarvitsee tehdä optimoinnin kuluessa tyypillisesti satoja tai tuhansia kertoja.

KANTAVIEN RAKENTEIDEN OPTIMOINTI

Kantavien rakenteiden optimoinnilla (structural optimization) tarkoitetaan hyvien lujuusopillisten rakennevaihtoehtojen systemaattista hakua optimoinnin tarjoamia keinoja hyväksikäyttäen. Tavoitteena on löytää sellaisia ratkaisuja, jotka perinteisellä suunnittelijan kokemukseen ja intuitioon perustuvalla menetelmällä olisivat muuten jääneet löytymättä. Intuitioon ja kokeiluihin perustuvassa parantelussa ei ole kyse optimoinnista, vaan yleensä tavallisesta tuotekehityksestä.

Kantavien rakenteiden optimoinnissa voidaan erottaa kolme luonteeltaan toisistaan poikkeavaa tehtävätyyppiä: mitoituksen, muodon ja topologian optimointi. Mitoitustehtävässä ongelmana on valita rakenteen poikkileikkausmitoille sellaiset arvot, että rakenteesta tulisi optimaalinen. Rakenteen tyyppi ja geometria ovat jo lyöty lukkoon ja tehtävänä on vain mitoittaa osat parhaalla mahdollisella tavalla. Esimerkiksi levymäisellä rakenteella tämä tarkoittaa valmistamiseen käytetyn levyn paksuuden valitsemista sopivasti.



Kuva 1. Levymäisen korvakkeen optimointi, kun kyseessä on a) mitoitustehtävä, b) muodon optimointi ja c) topologian optimointi.

Muodon optimoinnissa on puolestaan ideana muutella optimoitavan rakenteen muotoa. Pyritään siis valitsemaan sellainen muoto tai reunaviiva, että esimerkiksi materiaalin kulutus minimoituu ja rakenne vielä kestää. Muodon optimointi yhdistetään usein mitoitustehtävän kanssa, jolloin levymäisellä kappaleella muuteltaisiin reunaviivan kanssa samanaikaisesti myös levyn paksuutta.

Topologian optimoinnissa on kyse astetta pidemmälle viedystä optimoitavan kappaleen geometrian muuttelemisesta kuin muodon optimoinnissa. Nyt tarkoituksena on löytää paras mahdollinen ratkaisu vaikuttamalla mukaan tulevien rakenneosien määrään, tyyppiin ja sijoitteluun. Ristikolla voidaan valita optimiratkaisuun mukaan vain osa alun perin mahdollisista sauvoista tai levymäiseen kappaleeseen voidaan tehdä sopiva määrä reikiä. Topologian optimointi yhdistetään usein muodon optimoinnin ja mitoitustehtävän kanssa.

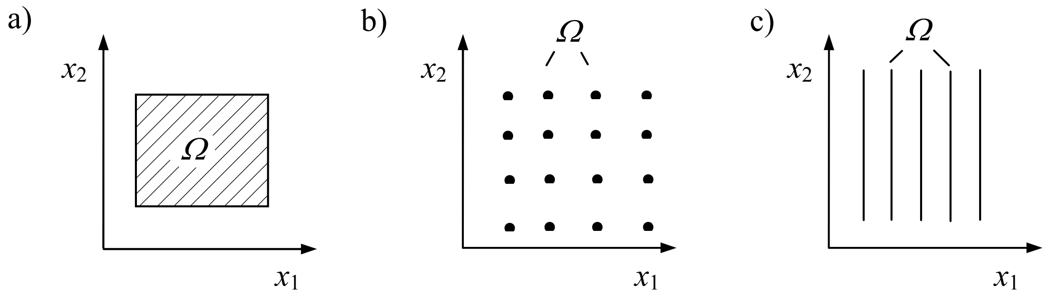
Optimointiongelman muodostamiseksi täytyy matemaattisesti määritellä suunnittelumuuttujista riippuva kohdefunktio, jonka lukuarvo ilmaisee minimoitavana tai maksimoitavana olevan suureen arvon. Tehtävässä voidaan lisäksi asettaa suunnittelumuuttujista riippuvia lisäehtoja, jotka täytyy pukea rajoitusyhtälöiksi ja rajoitusepäyhtälöiksi. Jos rajoitusehtoja ei ole, sanotaan optimointiongelmaa rajoittamattomaksi tai vapaaksi. Optimointiongelman ratkaiseminen tarkoittaa sellaisten arvojen määrittämistä suunnittelumuuttujille jossakin annetussa joukossa, että kohdefunktio saa pienimmän tai suurimman mahdollisen arvon niin, että rajoitusehdot toteutuvat.

Kantavien rakenteiden optimoinnissa on useita tyypillisiä vaihtoehtoja valita minimoitava tai maksimoitava kohdefunktio. Voidaan esimerkiksi ajatella, että kevein mahdollinen vaaditun kuormituksen kestävä rakenne on edullisin ja siten paras. Muita mahdollisia valintoja kohdefunktioksi ovat valmistuskustannukset, rakenteen jonkin pisteen siirtymä, luotettavuus tai vaikkapa alin ominaistaajuus. Kohdefunktioita voi olla myös kerrallaan useampia kuin yksi, jolloin puhutaan monitavoitteisesta optimointiongelmasta ja sen minimoitavista tai maksimoitavista kriteereistä. Kriteerit ovat keskenään ristiriitaisia eli monitavoitteisessa ongelmassa pyritään esimerkiksi yhtä aikaa mahdollisimman keveään ja jäykkään rakenteeseen.

Rajoitusehtojen tarkoituksena on huolehtia optimoidun rakenteen käyttökelpoisuudesta. Esimerkiksi massan minimointiongelmassa rakenteen keventyessä muodonmuutokset ja jännitykset kasvavat ja varmuus stabiilisuuden menettämisen suhteen pienenee. Jossakin vaiheessa saavutetaan tilanne, missä rakennetta ei voida enää keventää sen vaurioitumatta tai menettämättä toimivuuttaan tarkoitetulla tavalla. Siirtymä-, jännitys- ja stabiilisuusrajoitusten lisäksi kantavien rakenteiden optimoinnissa tavallisia rajoitusehtoja ovat ominaistaajuus-, luotettavuus-, tilavuus- ja symmetriarajoitukset.

Jos kohdefunktio ja rajoitusehtofunktiot ovat suunnittelumuuttujien suhteen lineaarisia, on optimointiongelma lineaarinen. Tavallisesti kantavien rakenteiden optimoinnissa päädytään kuitenkin epälineaariseen tehtävään, missä yleensä ainakin rajoitusehtofunktiot ovat epälineaarisia. Esimerkiksi staattisesti määräämättömällä ristikolla sauvojen normaalijännitykset eivät ole poikkipinta-aloista lineaarisesti riippuvia. Tällöin ristikon massaa minimoitaessa jännitysrajoitusehdoilla sauvojen pinta-alojen ollessa suunnittelumuuttujina on kohdefunktio lineaarinen, mutta rajoitusehtofunktiot epälineaarisia. Epälineaarinen optimointiongelma on pääsääntöisesti hankalampi ratkaista kuin lineaarinen tehtävä.

Optimointiongelmat voidaan jakaa suunnittelumuuttujien tyypin perusteella jatkuviin ja diskreetteihin ongelmiin sekä sekalukutehtäviin. Jatkuvassa optimointiongelmassa kukin suunnittelumuuttujista voi saada mitä tahansa arvoja joltakin sallitulta väliltä. Diskreetissä ongelmassa kaikki suunnittelumuuttujat voivat saada pelkästään joitakin tiettyjä arvoja, eivätkä mitään arvoa näiden välistä. Sekalukutehtävä on puolestaan näiden kahden edellisen yhdistelmä eli se sisältää sekä jatkuvia että diskreettejä muuttujia. Suunnittelumuuttujiin perustuva optimointiongelmien jaottelu on oleellista, koska sekalukutehtävät ja diskreetit ongelmat ovat usein huomattavasti työläämpiä ratkaista kuin pelkkiä jatkuvia muuttujia sisältävät ongelmat. Esimerkkinä jatkuvasta tehtävästä voidaan mainita levystä hitsatun I-palkin uuman korkeuden ja laipan leveyden valitseminen optimaalisesti. Diskreettejä muuttujia tarvitaan puolestaan tehtävässä, missä optimoidaan vaikkapa muovikomposiitin kuitukerrosten lukumäärää.



Kuva 2. Käypä joukko Ω, kun kyseessä on a) jatkuvat suunnittelumuuttujat, b) diskreetit suunnittelumuuttujat ja c) sekalukutehtävä.

Kantavien rakenteiden optimoinnin perusteisiin tutustuminen onnistuu oppikirjojen [2], [3], [6], [7] ja [9] avulla. Diskreettiä kantavien rakenteiden optimointia on puolestaan tarkasteltu kirjassa [5].

PARVEILUALGORITMI

Parveilualgoritmi voidaan luokitella ns. populaatiopohjaisiin heuristisiin optimointimenetelmiin, kuten myös sitä paljon laajemmalti käytetty geneettinen algoritmikin. Heuristiset menetelmät perustuvat usein johonkin luonnosta matkittuun päättelyyn, joka ei täytä ankaria loogisia vaatimuksia, mutta johtaa usein oikeaan tulokseen. On siis tavallaan kyse älykkäiden arvausten menetelmästä, joka optimoinnissa yleensä tuottaa hyvän lopputuloksen, muttei kuitenkaan välttämättä ongelman optimia. Heuristiset optimointimenetelmät ovat melko yksinkertaisia ja ne on helppo ohjelmoida tietokoneelle. Monet niistä ovat pienellä työllä muutettavissa eri sovellusalueiden ongelmien ratkaisuun sopiviksi. Heuristiset algoritmit ovat usein stokastisia, mutta eivät kuitenkaan välttämättä aina.

Ideana parveilualgoritmissa on matkia ruokaa tai pesäpaikkaa etsivän lintu-, hyönteis- tai kalaparven sosiaalista käyttäytymistä luonnossa. Parven jäsenet yrittävät sopeutua mahdollisimman hyvin ympäristöönsä käyttäen hyväksi omaa kokemustaan ja parven yhdessä keräämää tietoa ympäristöstä. Yksittäisen jäsenen, ja samalla myös koko parven, kannalta on hyödyllistä jakaa jäsenten tietoon tullut informaatio ympäristöstä kaikkien käyttöön parvessa. Parveilualgoritmin tutkimus on käynnistynyt 90-luvun puolivälissä ja sitä on aikaisemmin sovellettu mekaniikan ongelmiin artikkeleissa [4] ja [10].

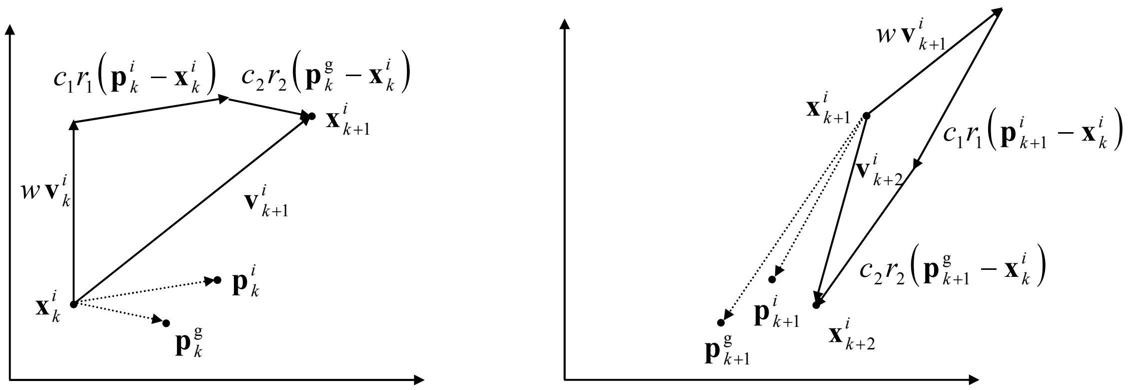
Parveilualgoritmissa jäsenen i uusi paikka iteraatiokierroksella k $\mathbf{x}_{k+1}^{i}$ riippuu sen nykyisestä paikasta $\mathbf{x}_{k}^{i}$ ja nopeudeksi nimitetystä termistä $\mathbf{v}_{k+1}^{i}$

$$\mathbf{x}_{k+1}^{i} = \mathbf{x}_{k}^{i} + \mathbf{v}_{k+1}^{i} \; . \tag{1}$$

Nopeus lasketaan kaavasta

$$\mathbf{v}_{k+1}^{i} = w\,\mathbf{v}_{k}^{i} + c_1 r_1\left(\mathbf{p}_{k}^{i} - \mathbf{x}_{k}^{i}\right) + c_2 r_2\left(\mathbf{p}_{k}^{\mathrm{g}} - \mathbf{x}_{k}^{i}\right) \; , \tag{2}$$

missä $\mathbf{p}_{k}^{i}$ on jäsenen i ja $\mathbf{p}_{k}^{\mathrm{g}}$ koko parven löytämä paras paikka iteraatiokierrokseen k mennessä. Tällöin siis ainakin yhden jäsenen paras paikka vastaa koko parven löytämää parasta paikkaa. Termiä w kutsutaan inertiaksi, r_1 ja r_2 ovat satunnaislukuja siten, että $r_1, r_2 \in [0,1]$, ja c_1 sekä c_2 ovat skaalausparametreja. Inertia w kontrolloi, kuinka laajasti hakuavaruutta tutkitaan optimointiprosessissa, ja tyypillinen valinta sen suuruudelle on $0{,}8 \le w \le 1{,}4$. Inertia voi olla tyypiltään dynaaminen, jolloin w:n arvo on aluksi suurempi ja pienenee optimoinnin kuluessa haun tarkentuessa kohti lupaavimpia alueita. Skaalausparametrit c_1 ja c_2 kontrolloivat, miten paljon parven jäsen i luottaa itseensä ja toisaalta parveen. Tyypillinen valinta niiden arvoiksi on $c_1 = c_2 = 2$.



a) kierros k a) kierros k+1

Kuva 3. Parveilualgoritmin iteraatioaskeleet a) pisteestä $\mathbf{x}_{k}^{i}$ pisteeseen $\mathbf{x}_{k+1}^{i}$ ja b) pisteestä $\mathbf{x}_{k+1}^{i}$ pisteeseen $\mathbf{x}_{k+2}^{i}$. Nopeuden (2) lausekkeessa inertiaan liittyvä termi $w\,\mathbf{v}_{k}^{i}$ suuntaa etsintää laajemmalle hakuavaruudessa ja pisteisiin $\mathbf{p}_{k}^{\mathrm{g}}$ ja $\mathbf{p}_{k}^{i}$ liittyvät termit kohti lupaaviksi osoittautuneita alueita.

Parveilualgoritmi on periaatteessa suunnittelumuuttujiltaan jatkuvien optimointitehtävien ratkaisumenetelmä. Diskreetit suunnittelumuuttujat voidaan huomioida yksinkertaisesti pyöristämällä muuttujien arvot aina lähimpään sallittuun arvoon. Rajoitusehdot käsitellään parveilualgoritmissa sakottamalla epäkäypiä ratkaisuita verrannollisesti niiden epäkäypyyteen nähden. Standardimuotoinen epäyhtälörajoitettu optimointiongelma

$$\begin{aligned} &\min f(\mathbf{x}) \\ &g_i(\mathbf{x}) \le 0 \qquad i = 1,\ldots,n \end{aligned} \tag{3}$$

korvataan uudella vapaalla ongelmalla

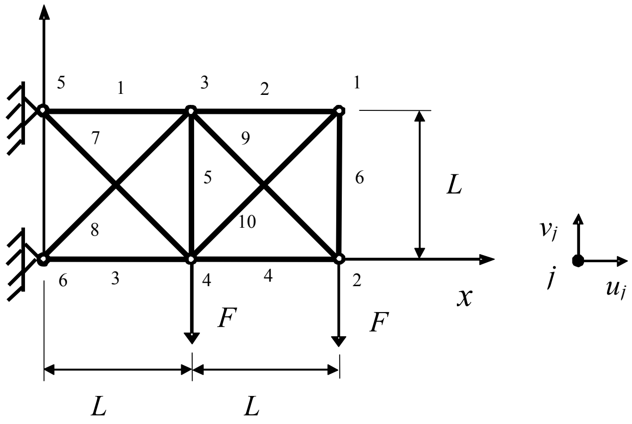
$$\min f(\mathbf{x})\left[1 + \sum_{g_i(\mathbf{x})>0} R\, g_i(\mathbf{x})\right], \tag{4}$$

missä R on sakkokerroin. Jos sakkokerroin valitaan liian pieneksi, tulevat turhan monet parven jäsenet epäkäyviksi, ja jos taas se valitaan liian suureksi, karsiutuvat hyvät ja vain lievästi epäkäyvät jäsenet pois. Sakkokertoimen suuruus määrää, miten paljon parveilualgoritmissa parven jäsenten hyvyyttä punnittaessa painotetaan kohdefunktion arvoa ja miten paljon käypyyttä. Sakotuksessa tulee pitää kirjaa parhaasta tiedetystä käyvästä ratkaisusta, koska muuten paras tiedetty ratkaisu $\mathbf{p}_k^{\mathrm{g}}$ voi olla epäkäypä.

Lopetuskriteerinä voidaan käyttää tiettyä kiinteää iteraatiokierrosten lukumäärää $k^{\max}$ tai vaihtoehtoisesti tarkkailla parhaan tiedetyn kohdefunktion arvon $f(\mathbf{p}_k^{\mathrm{g}})$ kehitystä ja lopettaa optimointi ellei se ole enää parantunut $\bar{k}^{\max}$ kierroksen kuluessa (Lopeta, jos $f(\mathbf{p}_k^{\mathrm{g}}) = f(\mathbf{p}_{k-\bar{k}^{\max}}^{\mathrm{g}})$ kierroksella k).

TASORISTIKON SAUVOJEN POIKKIPINTA-ALOJEN DISKREETTI OPTIMOINTI

Ensimmäinen esimerkkitehtävä käsittelee kuvan 4 kymmenen sauvan tasoristikon massan minimointia jännitys- ja siirtymärajoitusehdoilla. Ideana on valita suunnittelumuuttujina toimivien sauvojen poikkipinta-alat A_i annetusta äärellisestä joukosta niin, että rakenteen massa minimoituu ja asetetut rajoitusehdot toteutuvat. Jännitysrajoitusehdot rajoittavat kunkin sauvan normaalijännityksen alarajan $-\sigma^{\max} = -25\,000$ psi ja ylärajan $\sigma^{\max} = 25\,000$ psi väliin ja siirtymärajoitusehdot pakottavat kaikkien solmujen pystysuuntaisen liikkeen pienemmäksi kuin suurin sallittu arvo $v^{\max} = 2{,}0$ in. Ongelman suunnittelumuuttujat ovat tyypiltään diskreettejä, sillä niiden arvot valitaan 81 pinta-alan joukosta. Koska ristikon muotoa ja sauvojen lukumäärä ei muutella optimoinnin kuluessa ja haetaan vain optimaalisia sauvojen poikkipinta-aloja, on kyseessä kantavien rakenteiden optimoinnin mitoitustehtävä. Tarkasteltava optimointiongelma on otettu lähteestä [8].



$L = 360 \text{ in} = 9{,}1 \text{ m}$

$F = 10\,000 \text{ lb} = 44{,}5 \text{ kN}$

$E = 1{,}0 \cdot 10^7 \text{ psi} = 69{,}0 \text{ GPa}$

$\sigma^{\max} = 25\,000 \text{ psi} = 172{,}4 \text{ MPa}$

$v^{\max} = 2{,}0 \text{ in} = 50{,}8 \text{ mm}$

Kuva 4. Kymmenen sauvan tasoristikko.

Matemaattisesti esitettynä optimointiongelma on muotoa

$$\min m(\mathbf{x})$$

$$
\begin{aligned}
&-\sigma^{\max} \le \sigma_i(\mathbf{x}) \le \sigma^{\max} \qquad \forall\ i = 1,\ \ldots,\ 10 \\
&-v^{\max} \le v_j(\mathbf{x}) \le v^{\max} \qquad \forall\ j = 1,\ 2,\ 3,\ 4
\end{aligned} \tag{5}
$$

$$
\begin{aligned}
&A_i \in \{0{,}1\ ,\ 0{,}5\ ,\ 1{,}0\ ,\ 1{,}5\ ,\ 2{,}0\ , \ldots ,\ 39{,}5\ ,\ 40{,}0\}\ \text{in}^2 \\
&\mathbf{x} = [A_1\ A_2\ \ldots\ A_{10}]^{\mathrm{T}}
\end{aligned}
$$

Kaksoisepäyhtälörajoitukset voidaan kukin erikseen muuttaa kahdeksi standardimuodon mukaiseksi tavalliseksi epäyhtälörajoitukseksi

$$g_i(\mathbf{x}) = -\sigma^{\max} - \sigma_i(\mathbf{x}) \le 0 \qquad g_{i+10}(\mathbf{x}) = \sigma_i(\mathbf{x}) - \sigma^{\max} \le 0 \tag{6}$$

$$g_j(\mathbf{x}) = -v^{\max} - v_j(\mathbf{x}) \le 0 \qquad g_{j+4}(\mathbf{x}) = v_j(\mathbf{x}) - v^{\max} \le 0 \tag{7}$$

Epäyhtälörajoitusten lukumääräksi tulee siten 28. Sakotusta käytettäessä päädytään tarkastelemaan optimointiongelmaa

$$
\begin{aligned}
&\min m(\mathbf{x}) \left[1 + \sum_{\substack{g_i(\mathbf{x})>0 \\ i=1,\ldots,28}} R\, g_i(\mathbf{x}) \right] \\
&A_i \in \{0{,}1\ ,\ 0{,}5\ ,\ 1{,}0\ ,\ 1{,}5\ ,\ 2{,}0\ , \ldots ,\ 39{,}5\ ,\ 40{,}0\}\ \text{in}^2 \\
&\mathbf{x} = [A_1\ A_2\ \ldots\ A_{10}]^{\mathrm{T}}
\end{aligned} \tag{8}
$$

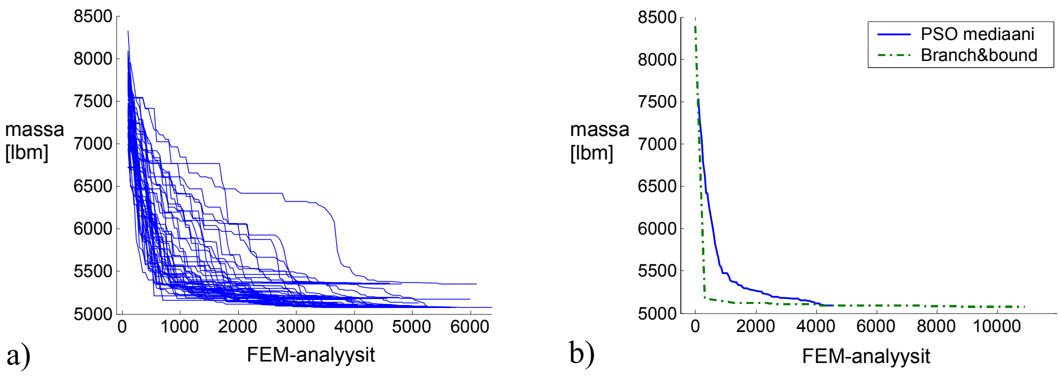
Pelkästään diskreettejä suunnittelumuuttujia sisältävässä optimointiongelmassa on eri ratkaisuvaihtoehtoja jokin äärellinen määrä. Niiden kaikkien läpikäyminen ja parhaan käyvän poimiminen eli totaalinen enumeraatio ei ole kuitenkaan käyttökelpoinen ratkaisumenetelmä vähänkään suuremmassa optimointiongelmassa. Kymmenen sauvan ristikon optimointiongelmassa, missä on 81 vaihtoehtoa kullekin sauvalle, on eri ratkaisuvaihtoehtojen lukumäärä $81^{10} \approx 1{,}216 \cdot 10^{19}$. Jos voitaisiin käydä läpi nopeata tietokonetta käyttäen 10^9 tapausta sekunnissa, kestäisi enumeraatio noin 386 vuotta.

Tasoristikon massan minimointiongelma on ratkaistu 50 kertaa peräjälkeen parveilualgoritmilla käyttäen kussakin optimointiajossa 50 yksilön kokoista parvea ja 200 iteraatiokierrosta. Alkuparvi on arvottu ja sakkokertoimen arvona on käytetty $R = 2$. Eri optimointiajoilla saatu massan muutos on esitetty kuvassa 5 a). Kuvassa 5 b) on verrattu parveilualgoritmin eri ajoista laskettua massan mediaanin kehitystä branch&bound –algoritmin antamaan tulokseen. Branch&bound on implisiittistä enumeraatiota käyttävä diskreettien optimointiongelmien ja sekalukutehtävien ratkaisualgoritmi, missä ratkaistaan joukko suunnittelumuuttujiltaan jatkuviksi relaksoituja optimointiongelmia. Jos relaksoidut ongelmat pystytään ratkaisemaan tarkasti, löytää Branch&bound varmasti kohtuullisen kokoisessa ongelmassa globaalin optimin.

	Sauvojen poikkipinta-alat [in^2]										
	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	massa [lbm]
PSO	31,0	0,1	24,0	15,0	0,1	0,1	8,5	20,0	21,5	0,1	5081,5
BB	29,5	0,1	24,0	16,0	0,1	0,5	7,5	20,0	22,5	0,1	5077,9
[7] 1)	31,5	0,1	23,0	15,5	0,1	0,5	7,5	20,5	21,0	0,1	5045,8

1) Artikkelin [8] optimina tarjoama ratkaisu ei ole käypä, vaan rikkoo solmun 1 siirtymärajoitusta.

Taulukko 1. Parveilualgoritmin (PSO) ja Branch&bound -algoritmin (BB) löytämät keveimmät rakenteet sekä artikkelissa [8] esitetyt pinta-alat kymmenen sauvan ristikon optimointiongelmassa.

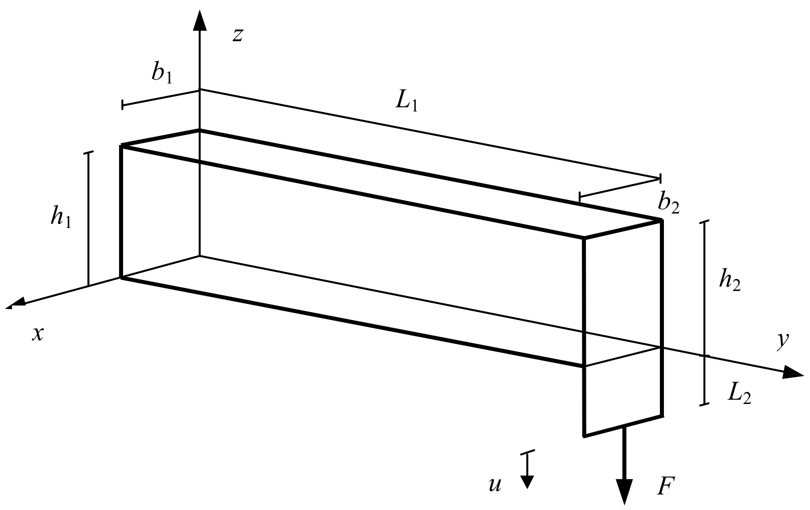


Kuva 5. a) Kymmenen sauvan ristikon optimointiongelman kohdefunktion kehitys 50 perättäisellä parveilualgoritmin ajolla laskentaan kuluneiden FEM-analyysien funktiona. b) Kohdefunktion mediaani a)-kohdan optimointiajoista laskettuna ja Branch&bound -algoritmin antama tulos.

Kuvasta 5 a) nähdään selvästi stokastisuuden vaikutus parveilualgoritmin tuloksissa. Joillakin optimointiajoilla saatetaan löytää kevyt rakenne nopeasti, mutta vastaavasti taas toisilla massan pieneneminen on melko hidasta. Kuvan 5 b) perusteella parveilualgoritmi näyttäisi olevan (keskimäärin) kilpailukykyinen Branch&bound -algoritmin kanssa. Branch&bound tuottaa hieman nopeammin lähellä lopullista ratkaisua olevan tuloksen, mutta sen vaatima FEM-analyysien kokonaismäärä on suurempi kuin mitä parveilualgoritmin mediaanituloksen käyttämä määrä on. Merkittävää on, että molemmat menetelmät vaativat hyvin monta FEM-analyysiä. Taulukon 1 perusteella parveilualgoritmi ei löydä yhdelläkään 50 ajosta niin hyvää tulosta kuin mitä saadaan deterministisen Branch&bound -algoritmin ainoalla ajolla.

TERÄSKOTELON YHDISTETTY MUODON JA MITOITUKSEN OPTIMOINTI

Toisessa esimerkkitehtävässä tarkastellaan kuvan 6 mukaista teräslevystä hitsaamalla koottua ja toisesta päästä tuettua koteloa, jonka kuormituksena on pistevoima. Tehtävänä on valita kotelon muotoa kuvaaville suunnittelumuuttujille h_1, b_1, h_2 ja b_2 sekä levynpaksuuksille t_1 (pohja), t_2 (sivulevyt), t_3 (yläpinta) ja t_4 (pääty) sellaiset arvot, että rakenteesta tulee mahdollisimman kevyt. Optimoidulla kotelolla pään pystysiirtymä u ei saa ylittää arvoa $u^{\max} = 5\ \mathrm{mm}$, von Misesin mukainen vertailujännitys ei saa ylittää arvoa $\sigma^{\max} = 237\ \mathrm{MPa}$ missään kohdin rakennetta ja kotelon lineaarisen stabiilisuuteorian mukainen kuormituskerroin λ_{kr} täytyy olla vähintään $n_{\mathrm{kr}} = 3$. Suunnittelumuuttujat h_1, b_1, h_2 ja b_2 ovat tyypiltään jatkuvia ja niille kullekin on asetettu kiinteät ala- ja ylärajat ($h_1^{\min} = h_2^{\min} = 0{,}1\ \mathrm{m}$, $h_1^{\max} = h_2^{\max} = 0{,}5\ \mathrm{m}$, $b_1^{\min} = b_2^{\min} = 0{,}05\ \mathrm{m}$ ja $b_1^{\max} = b_2^{\max} = 0{,}2\ \mathrm{m}$). Levyn paksuudet t_1, t_2, t_3 ja t_4 ovat puolestaan diskreettejä suunnittelumuuttujia, joiden arvot tulee valita joukosta {3, 5, 7, 10, 12, 15, 20, 25, 30} mm. Täten tarkasteltava yhdistetty muodon ja mitoituksen optimointiongelma on tyypiltään sekalukutehtävä.



$F = 50\ \mathrm{kN}$
$L_1 = 2{,}0\ \mathrm{m}$
$L_2 = 0{,}2\ \mathrm{m}$
$E = 210\ \mathrm{GPa}$
$\nu = 0{,}3$
$\rho = 7850\ \dfrac{\mathrm{kg}}{\mathrm{m}^3}$

Kuva 6. Hitsatun teräskotelon massan minimointi muotoa ja levyjen paksuuksia muuttelemalla. Kotelon pohja pysyy optimoinnissa *xy*-tasolla.

Matemaattisesti esitettynä optimointiongelma on

$$
\begin{aligned}
&\min m(\mathbf{x}) \\
\\
&u(\mathbf{x}) \le u^{\max} \\
&\sigma_{\text{mises}}(\mathbf{x}) \le \sigma^{\max} \\
&\lambda_{\text{kr}}(\mathbf{x}) \ge n_{\text{kr}} \\
\\
&h_1^{\min} \le h_1 \le h_1^{\max} \\
&b_1^{\min} \le b_1 \le b_1^{\max} \\
&h_2^{\min} \le h_2 \le h_2^{\max} \\
&b_2^{\min} \le b_2 \le b_2^{\max} \\
&t_1, t_2, t_3, t_4 \in \left\{ t^1, t^2, \ldots, t^n \right\} \\
\\
&\mathbf{x} = \left[h_1 \; b_1 \; h_2 \; b_2 \; t_1 \; t_2 \; t_3 \; t_4 \right]^{\mathrm{T}}
\end{aligned}
\tag{9}
$$

Sakotuksessa epäyhtälörajoitukset muutetaan muotoon $g_1(\mathbf{x}) = u(\mathbf{x}) - u^{\max} \le 0$, $g_2(\mathbf{x}) = \sigma_{\text{mises}}(\mathbf{x}) - \sigma^{\max} \le 0$ ja $g_3(\mathbf{x}) = n_{\text{kr}} - \lambda_{\text{kr}}(\mathbf{x}) \le 0$ ja kohdefunktioksi otetaan

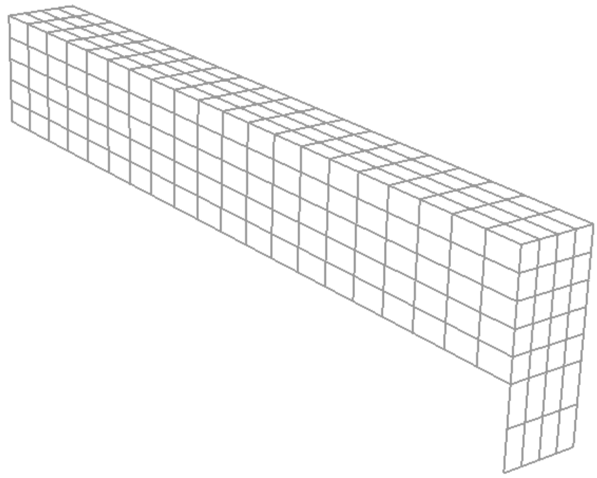
$$
\min m(\mathbf{x}) \left[1 + \sum_{\substack{g_i(\mathbf{x})>0 \\ i=1,2,3}} R\, g_i(\mathbf{x}) \right]. \tag{10}
$$

Muuttujarajoitukset säilyvät entisellään.

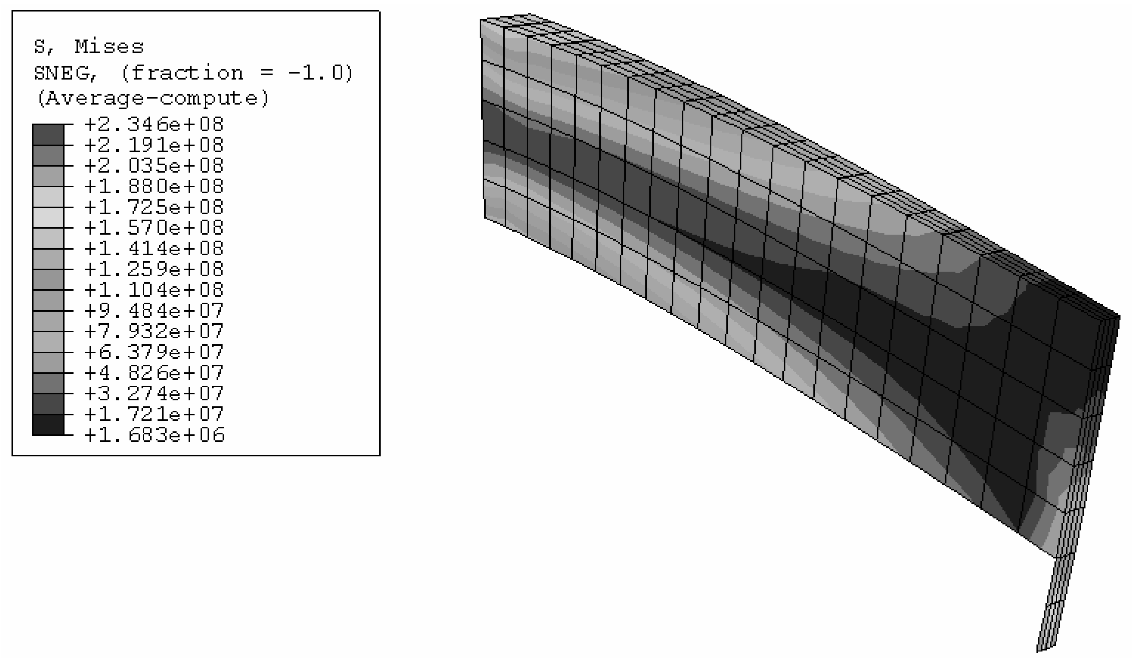
Kotelon analysointi tehdään käyttäen ABAQUSta [1] laskentamallilla, jossa on 388 kpl S4-tyyppisiä kuorielementtejä. Kuvassa 7 on esitetty laskuissa käytetyn mallin elementtiverkko.

Kotelon optimointiongelma on ratkaistu kymmenen kertaa parveilualgoritmilla käyttäen 15 yksilön parvea, 50 iteraatiokierrosta, arvottua alkuparvea ja rajoitusehtojen sakotusta. Löydetyn keveimmän rakenteen suunnittelumuuttujien ja kohdefunktion sekä pään siirtymän u, vertailujännityksen σ_{mises} ja kuormituskertoimen λ_{kr} arvot on esitetty taulukossa 2. Kuvasta 8 selviää vastaavan rakenteen deformoitunut muoto ja vertailujännitysjakauma.

Kuvan 8 perusteella nähdään, että optimoidun kotelon muoto on kapea ja korkea. Tämä tuntuukin melko luontevalta tulokselta palkkiteorian kannalta asiaa mietittäessä. Kotelon sivujen paksuus t_2 vaikuttaa pieneltä sen korkeuteen nähden, mutta toisaalta myös taivutusjännitys on melko matala, kuten kuvasta 8 voidaan päätellä.



Kuva 7. Hitsatun kotelon analysointiin käytetty elementtiverkko.



Kuva 8. Keveimmän löydetyn kotelon deformoitunut muoto ja von Misesin vertailujännityksen jakauma. Suurin vertailujännitys esiintyy pistevoiman vaikutus-kohdassa ja muualla kotelossa sen arvo on selkeästi matalampi.

Taulukosta 2 nähdään, että kaikki rajoitusehdot ovat melko lähellä suurinta tai pienintä sallittua arvoansa. Lineaarisen stabiilisuusteorian kannalta FEM-mallin verkon tiheys olisi hyvä olla suurempi kotelon sivuissa, jotta saataisiin luotettavampia tuloksia kuormitus-kertoimelle. Keveimmän rakenteen antanut optimointiajo vaati 812 FEM-analyysin tekemisen.

massa	u	σ_{mises}	λ_{kr}				
66,12 kg	4,99 mm	235,3 MPa	3,33				
h_1	b_1	h_2	b_2	t_1	t_2	t_3	t_4
500,0	142,3	479,4	50,0	7	3	5	7

Taulukko 2. Keveimmän parveilualgoritmilla löydetyn kotelon suunnittelumuuttujien arvot, massa, pään pystysiirtymä, suurin von Mises vertailujännitys ja lineaarisen stabiilisuuteorian mukainen kuormituskerroin.

YHTEENVETO

Voidaan todeta, että optimoinnille on selkeä tarve lujuusopillisessa mitoituksessa. Kantavien rakenteiden optimoinnin merkitys tulee tulevaisuudessa kasvamaan, koska se pystyy tarjoamaan selkeän keinon saavuttaa etua totuttuihin suunnittelumenetelmiin nähden. Tietokoneiden kasvava laskentakapasiteetti ja kehittyvät optimointialgoritmit mahdollistavat yhä suurempien ongelmien ratkaisemisen kohtuullisessa ajassa.

Parveilualgoritmi näyttää esitettyjen esimerkkien valossa toimivan kohtuullisen hyvin. Sillä on monia miellyttäviä ominaisuuksia, mutta myös joitakin selkeitä puutteita. Tärkeimpänä etuna voidaan mainita menetelmän yksinkertaisuus ja helppo sovellettavuus erilaisiin optimointiongelmiin. Lisäksi rakenteen analysointi voidaan tehdä valmiilla kaupallisella FEM-ohjelmalla, mikä on tärkeää monissa käytännön sovelluksissa. Parveilualgoritmi ei ole myöskään niin herkkä suunnittelumuuttujien tai rajoitusehtojen lukumäärille ja tyypille kuin mitä monet muut optimointialgoritmit ovat.

Parveilualgoritmin suurimpana puutteena on optimoinnissa kuluvien rakenneanalyysien melko suuri määrä. Optimoitavan rakenteen FEM-malli ei voi olla kovinkaan iso tai muuten laskennallisesti raskas. Myös parveilualgoritmin stokastisuus vaikeuttaa tulosten luotettavuuden arviointia. Jotta voitaisiin olla varmoja löydetyn lopputuloksen hyvyydestä, tulee tehdä useampia optimointiajoja. Muutenkin parveilualgoritmin käyttö uudessa tehtävässä vaatii yleensä aina sen eri parametrien arvojen virittämistä kokeilemalla.

LÄHTEET

[1] ABAQUS Student Edition Version 6.4

[2] Arora J. S. 1989. *Introduction to Optimum Design*. McGraw-Hill.

[3] Farkas J. & Jarmai K. 1997. *Analysis and Optimum Design of Metal Structures*. A. A. Balkama.

[4] Fourie P.C. &. Groenwold A. A. 2002. The particle swarm optimization algorithm in size and shape optimization. *Structural and Multidisciplinary Optimization*, 23, 259-267.

[5] Gutkowski W. (ed.) 1997. *Discrete Structural Optimization*. Springer-Verlag.

[6] Haftka R. & Gürdal Z. 1992. *Elements of Structural Optimization*. Kluwer.

[7] Kirsch U. 1993. *Structural Optimization*. Spinger-Verlag.

[8] Olsen G. & Vanderplaats G. 1989. Method for Nonlinear Optimization with Discrete Design Variables. *AIAA Journal*, Vol. 27, No 11, 1584-1589.

[9] Vanderplaats G. 1999. *Numerical Optimization Techniques for Engineering Design*. Vanderplaats Research & Development.

[10] G. Venter, & J. Sobieszczanski-Sobieski. 2003. Particle Swarm Optimization. *AIAA Journal*, Vol. 41, No 8, 1583-1589.

Jussi Jalkanen, yliassistentti

Tampereen teknillinen yliopisto

Teknillisen mekaniikan ja optimoinnin laitos

PL 589

33101 Tampere

jussi.jalkanen@tut.fi