

Jaakko Huusko

TERÄSRAKENTEIDEN DISKREETTI OPTIMOINTI

Rakennetun ympäristön tiedekunta
Diplomityö
Huhtikuu 2021

TIIVISTELMÄ

Jaakko Huusko: Teräsrakenteiden diskreetti optimointi
Diplomityö
Tampereen yliopisto
Rakennustekniikan diplomi-insimöörin tutkinto-ohjelma
Huhtikuu 2021

Diskreetillä optimoinnilla tarkoitetaan optimointitehtävää, missä yksi tai useampi suunnittelu-
muuttujista saa diskreettejä arvoja, kuten kokonaislukuja tai binäärisiä arvoja. Tavanomaiset teräs-
rakenteet muodostuvat teollisuudessa yleisesti käytetystä standardiprofiileista, joka tekee suunnit-
telutehtävästä diskreetin. Diskreettien optimointitehtävien haasteena on hakuavaruuden laajuus,
sillä sen suuruus kasvaa eksponentiaalisesti käytettävissä olevien profiilien mukaan.

Työssä tarkastellaan erilaisia menetelmiä lähestyä diskreettiä optimointiongelmia käyttäen
apuna kirjallisuudessa yleisesti käytettyjä ristikkorakenteita. Työn teoriaosuudessa käydään läpi
matemaattisen optimoinnin käsitteitä sekä tehtävien ratkaisemiseen käytettyjen algoritmien toi-
mintaperiaatteet.

Työssä tarkastellaan eurokoodin mukaisesti mitoitettun tasoristikon geometriaoptimointia. Lii-
tosten mitoitus suoritetaan geometriaoptimoinnin kanssa samanaikaisesti. Eurokoodin mukaiset
sauvan ja liitoksen kestävyys ehdot esitetään muodostettaessa optimointitehtävän rajoitusehtoja.
Optimointitehtävän kohdefunktiona käytetään ristikon massaa.

Ristikko optimoidaan kahdessa vaiheessa. Ensimmäisessä vaiheessa tehtävä relaxoidaan
ja ratkaistaan jatkuvien muuttujien avulla. Toisessa vaiheessa ratkaisu diskretoidaan käyttäen in-
deksimuuttujia. Optimoidun ristikon massa jakautuu 50-30-20 -suhteessa ylä- ja alapaarten sekä
uumasauvojen kesken. Jatkuvan tehtävän ratkaisu on noin 10 % kevyempi kuin paras diskreetti
ratkaisu. Määrääviksi rajoitusehdoiksi muodostuu lähes kaikissa tapauksissa jokin liitoksen geo-
metriaehdoista tai yläpaarten stabiilisuus.

Tehtävien ratkaisemiseen käytettiin Python -ohjelmointikielellä kehitettyä optimointikehystä. Al-
goritmien implementoimisessa hyödynnettiin DEAP (Distributed Evolutionary Algorithms in Pyt-
hon) sekä COIN-OR (Computational Infrastructure for Operations Research) julkaisemia avoimen
lähdekoodin ratkaisijoita.

Avainsanat: teräsrakenteet, diskreetti optimointi, ristikko, massan minimointi

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

ABSTRACT

Jaakko Huusko: Discrete Optimization of Steel Structures
Master's Thesis
Tampere University
Master's Degree Programme in Civil Engineering
April 2021

Discrete optimization is a branch of mathematical optimization, where some or all of the design variables get discrete values, such as integers or binary values. Common steel structures are assembled using standardized profiles which makes the optimization problem discrete. The challenge of discrete problems is the size of the search space, which grows exponentially with respect to the size of the available profiles.

Different methods for approaching discrete optimization problems are discussed using commonly used benchmark structures for comparing the results. Common terminology of mathematical optimization and the algorithms used in this study are expressed in the theory section of this study.

Geometry optimization of one-span planar steel truss according to Eurocodes is discussed. Truss' joints are designed simultaneously during geometry optimization. Steel member design and joint strength formulas according to Eurocodes are represented while formulating optimization problem's constraints. The weight of the truss is used as the optimization problem's objective function.

A two-stage optimization process is used to solve the discrete truss optimization problem. In the first phase the problem is relaxed and continuous variables are used to solve the problem. The second phase consist of discretizing the continuous solution using index variables. Continuous solution is approximately 10 % lighter than the best discrete solution. In most cases joint geometry or member stability constraints define the optimal solution.

Problems are solved using optimization framework developed using Python programming language. Optimization solvers are implemented using DEAP (Distributed Evolutionary Algorithms in Python) and COIN-OR (Computational Infrastructure for Operations Research) open source optimization libraries.

Keywords: steel structures, discrete optimization, plane truss, weight minimization

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Tämä työ on laadittu yhteistyössä Tampereen Yliopiston kanssa. Haluan kiittää etenkin Kristo Melaa, joka herätti kiinnostukseni teräsrakenteiden optimointia kohtaan jo kandidaatintyöni aikana.

Haluan kiittää myös opiskelijakavereitani Eeroa, Villeä, Tommia, Esaa sekä JV:tä. He tekivät opiskeluaikastani todella mieltuisaa. Kiitos myös perheelle ja ystäville.

Tampereella, 21. huhtikuuta 2021

Jaakko Huusko

SISÄLLYSLUETTELO

1	Johdanto	1
1.1	Tutkimuksen tausta	1
1.2	Tutkimuksen tavoitteet ja toteutus	2
2	Yleistä optimoinnista	4
2.1	Rajoitettu optimointi	5
2.1.1	Sakkofunktio	6
2.2	Konveksisuus	7
2.3	Suunnittelumuuttujat	7
2.4	Optimoinnin toteutus	8
3	Diskreetin optimoinnin menetelmiä	10
3.1	Yhden muuttujan approksiomenetelmät	10
3.2	Yhden muuttujan menetelmät ilman approksimaatiota	11
3.3	Usean muuttujan menetelmät	12
3.4	Pyöritysmenetelmät	13
4	Tutkimuksessa käytettävät algoritmit	16
4.1	Deterministiset algoritmit	16
4.1.1	Peräkkäisten linearisointien menetelmä, SLP	16
4.1.2	Peräkkäinen sekalukutehtävän linearisointi, MISLP	18
4.2	Metaheuristiset algoritmit	19
4.2.1	Muuttuvan ympäristön haku, VNS	20
4.2.2	Geneettinen algoritmi, GA	22
5	Diskreetin optimoinnin verrokitehtäviä	26
5.1	10-sauvan ristikko	26
5.2	15-sauvan ristikko	29
5.3	52-sauvan ristikko	32
5.4	Tulosten yhteenveto	35
6	Ristikon optimointi	37
6.1	Tehtävän formulointi	37
6.1.1	Suunnittelumuuttujat	38
6.1.2	Ristikon rakennemalli	39
6.2	Rajoitusehdot	40
6.2.1	Poikkipintasuureiden määrittäminen	40
6.2.2	Poikkileikkauksen kestävyys	42
6.2.3	Sauvan stabiilisuus	45
6.2.4	Liitoksen geometria	47
6.2.5	Liitoksen kestävyys	49

6.3 Tulokset	52
6.3.1 Ristikon geometriaoptimointi	53
6.3.2 Uumasauvojen nurjahduspituuden vaikutus	57
6.3.3 36m ristikon geometriaoptimointi	57
6.3.4 Huomioita ristikon optimoinnista	60
6.3.5 Liitosmitoituksen haasteet	61
7 Yhteenveto	64
Lähteet	67
Liite A Optimoinnissa käytetyt diskreetit poikkileikkaukset	70

KUVALUETTELO

1.1	Esimerkki optimoitavasta rakenteesta	2
2.1	Optimointiongelmien jaottelu Lin et. al mukaan [29]	4
2.2	Epäkonveksissa funktiossa vai olla monta lokaalia minimiä	7
2.3	Indeksimuuttujan saadessa arvon 3, valikoituu profiiliksi IPE 160	8
2.4	Jatkuvan muuttujan arvon muodostuminen binäärimuuttujien avulla. Muuttujan arvo ei ole kuitenkaan käypä, sillä binäärimuuttujien summa on 2	8
2.5	Optimoinnin kulku	9
3.1	RHS -profiileille on vaikea löytää sovitetta jolla saisi tarkasti approksimoitua poikkileikkauksen ominaisuuksia	11
3.2	I-poikkileikkauksien korkeudelle ja leveydelle muodostettu konvekssi verho	13
3.3	Toisessa vaiheessa käytettävät diskreetit arvot voidaan valita jatkuvan ratkaisun ympäriltä esimerkiksi euklidisen etäisyyden perusteella	13
3.4	Branch and Bound -menetelmän toimintaperiaate	14
3.5	Kandidaattijoukon muodostaminen jatkuvan ratkaisun lähimmän diskreetin arvon ympäriltä	15
4.1	Pienennyskertoimen γ :n vaikutus muuttujarajojen arvoon iteroinnin edetessä	18
4.2	Hakuympäristön muutostavat visualisoituna	21
4.3	Geneettisen algoritmin vuokaavio ja eri vaiheessa pohdittavia kysymyksiä	22
4.4	Populaatio muodostuu kromosomeista, jotka muodostuvat geeneistä	23
4.5	Risteytysmenetelmiä havainnollistettuna	24
4.6	Mutaatio vaikuttaa kromosomin yksittäisiin geeneihin	24
4.7	Rulettimenetelmässä vanhemmat valitaan simuloidun rulettipyöräytyksen tuloksena	25
5.1	10-sauvan ristikko ja tehtävässä käytettyjen parametrien arvot	27
5.2	Optimitulokseen päästään SLP-algoritilla lähes joka kerta satunnaisesta aloituspisteestä. Vain suurilla muuttujarajan arvoilla algoritmi saattaa jäädä jumiin epäkäypään lokaaliin minimiin.	28
5.3	10-sauvan ristikon optimiratkaisu	29
5.4	15-sauvan ristikko ja tehtävässä käytettyjen parametrien arvot	29
5.5	15-sauvan ristikon optimiratkaisu	32
5.6	52-sauvan ristikko ja tehtävässä käytettyjen parametrien arvot	33
6.1	Ristikon alkumitat sekä liitosten ja sauvojen numerointi	38
6.2	Neliöputkiprofiilin dimensiot ja plastisen taivutusvastuksen määrittäminen	41
6.3	Sauvojen systeempituudet ja yläpaarteen elementtijako	46

6.4	K-liitoksessa pisteiden c01 ja c1 sekä c02 ja c2 välille muodostetaan jäykät epäkeskisyysselementit	48
6.5	Suurilla kaltevuuskulmilla alapaarteen liitoksen epäkeskisyys ei pysy sallituissa rajoissa	54
6.6	Jatkuvan tehtävän kohdefunktio ja rajoitusehtojen maksimiarvo iteroinnin edetessä	54
6.7	Kattoristikon optimigeometria (yhtenäinen viiva) verrattuna alkuperäiseen geometriaan (katkoviiva)	55
6.8	Ristikon massan jakautuminen eri sauvojen kesken	57
6.9	36 metrin ristikon optimigeometria verrattuna alkuperäiseen geometriaan	59
6.10	Eri parametreilla indeksimuuttujan alilista muodostuu eri profiileista	62
6.11	Uumasauvan profiilin muutos vaikuttaa liitoksen vapaaväliin ja epäkeskisyyteen	62
6.12	Paarteen seinämäpaksuuden ja leveyden vaikutus paarteen pinnan kestävyteen	63

TAULUKKOLUETTELO

5.1	10-sauvan ristikon optimitulokset	28
5.2	15-sauvan ristikon geometriamuuttujien rajat	30
5.3	15-sauvan ristikon optimitulokset	31
5.4	52-sauvan ristikon optimitulokset	34
6.1	S355J2H teräksen materiaaliominaisuudet	39
6.2	Diskreetit geometriaoptimoinnin tulokset	56
6.3	24m ristikon geometriaoptimoinnin tulosten vertailu	56
6.4	Uumasauvan nurjahduspituuden vaikutus optimitulokseen	58
6.5	Nurjahduspituuden vaikutus diskreettiin tulokseen	58
6.6	Diskreetit 36m geometriaoptimoinnin tulokset	59
6.7	36m ristikon geometriaoptimoinnin tulosten vertailu	60
6.8	24m ristikon optimitulokset VNS:n avulla	61
A.1	10-sauvan optimoinnissa käytettävät pinta-alat, A_{10}	70
A.2	15-sauvan optimoinnissa käytettävät pinta-alat, A_{15}	70
A.3	52-sauvan optimoinnissa käytettävät pinta-alat, A_{52}	71
A.4	Ristikon optimoinnissa käytetty SHS-profiilikatalogi	72

1 JOHDANTO

1.1 Tutkimuksen tausta

Rakennesuunnittelijat haastetaan suunnittelemaan rakenteita jotka käyttävät mahdollisimman vähän resursseja (aikaa, rahaa, materiaalia), samalla täyttäen rakenteelle asetetut standardin mukaiset vaatimukset. Optimoinnin tarkoituksena on automatisoida erilaisten rakenneratkaisujen tarkastelua ja helpottaa näin rakennesuunnittelijan päätöksentekoa. Kaikki tehtävät, missä tiettyjen parametrien täytyy täyttää niille asetetut ehdot, voidaan ratkaista optimoinnin avulla. [1]

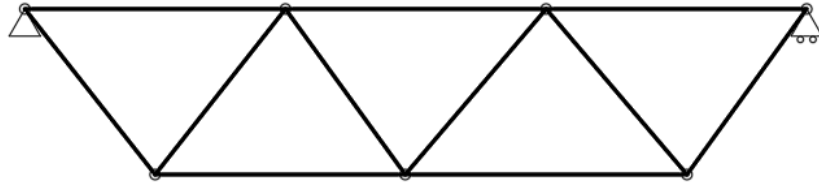
Diskreetillä optimoinnilla tarkoitetaan optimointitehtävää, jossa yksi tai useampi suunnittelumuuttujista saa vain diskreettejä arvoja, kuten kokonaislukuja tai binäärisiä arvoja. Optimointiongelmaa, jossa suunnittelumuuttujina käytetään sekä diskreettejä, että jatkuvia muuttujia kutsutaan sekamuuttujatehtäväksi (Mixed Variable Optimization) [11]. Esimerkki tämän tyyppisestä tehtävästä on ristikon geometrian optimointi, jossa sauvoille voidaan käyttää diskreettejä muuttujia ja solmusijainneille jatkuvia.

Rakenteiden optimointi jaetaan tyypillisesti kolmeen eri kategoriaan: mitoitus-, geometria- ja topologiaoptimointiin. Mitoitusoptimoinnissa tavoitteena on löytää optimipoikkileikkaukset joiden avulla rakenne on kohdefunktion suhteen optimaalisin. Poikkileikkausten muuttuessa rasitukset jakautuvat sauvojen kesken uudelleen, joten rakenteesta voidaan löytää sauvoja joiden poikkileikkauksen kasvattamisella saadaan suurin hyöty optimoinnin kannalta. Tyypillinen mitoitusoptimoinnin tehtävä on esimerkiksi Krajanvan [26] julkaisun mukaisen teräskehän optimointi.

Geometriaoptimoinnissa suunnittelumuuttujina käytetään poikkileikkausmuuttujien lisäksi myös solmupisteiden koordinaatteja. Koordinaatit voidaan esittää joko sauvojen lokaalissa koordinaatistossa tai tehtävän koordinaatistossa. Solmuja siirtämällä pystytään vaikuttamaan rakenteen rasitusten jakautumisen lisäksi myös yksittäisten sauvojen nurjahduspituuksiin. Geometriaoptimointi onkin yleistä rakenteissa, missä on pitkiä puristettuja sauvoja, kuten teräsristikko.

Rakenteen tarkempaa optimimuotoa haettaessa tulee kyseeseen topologiaoptimointi. Tässä tehtävätyypissä rakenteeseen voidaan lisätä ja poistaa sauvoja. Topologiaoptimoinnin avulla voidaan ratkaista esimerkiksi onko ristikkoon järkevä lisätä vertikaalisauvoja vai ei [19]. Tässä työssä tarkastellaan vain mitoitus- ja geometriaoptimointia.

Suunnittelija valitsee teräsrakenteissa käytettävät profiilit yleensä diskreetteistä profiilikatalogeista, jotta suunnittelua saadaan yksinkertaistettua ja välttämään ylimääräisiltä kustannuksilta, jotka aiheutuvat mielivaltaisen poikkileikkauksen valmistuksesta. Tämän tyyppinen suunnittelu saattaa vaikuttaa yksinkertaiselta, valitaan vaan kevyimmät profiilit, jotka täyttävät suunnittelustandardissa esitetyt rakenteelliset vaatimukset. Todellisuudessa ongelma ei kuitenkaan ole näin yksinkertainen kuin saattaa ensisilmäyksellä vaikuttaa.



Kuva 1.1. Esimerkki optimoitavasta rakenteesta

Käytettäessä diskreettejä profiilikatalogeja, on suunnitteluongelma diskreetti linkitetty ongelma, jossa rakenneosan ominaisuudet riippuvat valitusta profiilista. Kyseessä on kombinatorinen ongelma, jossa eri ratkaisujen lukumäärä kasvaa eksponentiaalisesti rakenneosien lukumäärän kasvaessa. Tarkastellaan kuvan 1.1 mukaista 10 sauvan ristikkorakennetta, jossa suunnittelijalla on käytettävissään 20 erilaista profiilia. Mahdollisia ratkaisuja on yhteensä yli 10 biljoonaa vaihtoehtoa, joten yhden ratkaisun tarkastelun kestäessä yhden millisekunnin, kestäisi kaikkien vaihtoehtojen läpikäyminen yli 300 vuotta.

1.2 Tutkimuksen tavoitteet ja toteutus

Työn tarkoituksena on perehtyä erilaisiin menetelmiin käsitellä diskreettiä teräsrakenteiden optimointiongelmaa. Esimerkkitehtävinä eri menetelmien vertailuun ja diskreettien tehtävien haastavuuden kuvaamiseen käytetään kirjallisuudessa yleisesti käytettyjä yksinkertaisia sauvarakenteita sekä eurokoodin mukaisesti mitoitettua tasoristikkoa. Tasoristikon optimoinnin yhteydessä käydään läpi eurokoodin mukainen sauvan sekä sauvojen välisen liitoksen kestävyuden laskenta.

Vaikka teräsrakenteiden diskreettiä optimointia on tutkittu jo monia vuosikymmeniä [33], uusimmissa tieteellisissä julkaisuissa keskitytään tehtävän muodostamisen sijaan optimaal algoritmien kehittämiseen. Julkaisuissa käytetyt rakenteet on myös hyvin usein yksinkertaisia sauvarakenteita, joita käytetään myös tässä tutkimuksessa algoritmien vertailuun luvussa 5. Standardin mukaisesti mitoitettujen teräsristikoiden optimoinnista löytyy todella vähän julkaisuja ja löydettyissä julkaisuissa liitosten mitoitusta ei ole huomioitu.

Tässä työssä tarkastellaan teräsristikon mitoitusta kahdessa vaiheessa. Ensimmäisessä vaiheessa ratkaistaan relaxoitu tehtävä jatkuvien muuttujien avulla ja toisessa vaiheessa ratkaisu diskretoidaan indeksimuuttujien avulla. Liitosten kestävyys- ja geometriaehdot tarkastetaan optimoinnin aikana. Ristikon profiileina käytetään SSAB:n neliöputkiprofiileja.

Työssä tarkasteltavat tutkimuskysymykset:

- Millä eri tavoilla disrkeetti optimointitehtävä pystytään esittämään?
- Miten valittu optimointialgoritmi vaikuttaa tehtävän ratkaisemiseen?
- Millaisia haasteita diskreetit muuttujat tuovat tehtävälle?

Kaikki tässä työssä esitettävät tehtävät on ratkaistu kannettavan tietokoneen kaksiytimisellä Intel i5-7200U @2.5 GHz prosessorilla ja 8GB RAM-muistilla. Optimointiin ja FEM-laskentaan käytettävä ohjelma on kirjoitettu Python -ohjelmointikielellä ja toteutettu yhteistyössä Kristo Melan kanssa. Itse ohjelmaa ja sen toimintaa ei ole optimoitu, joten optimointiin kuuluva laskenta-aika on suuntaa antava.

2 YLEISTÄ OPTIMOINNISTA

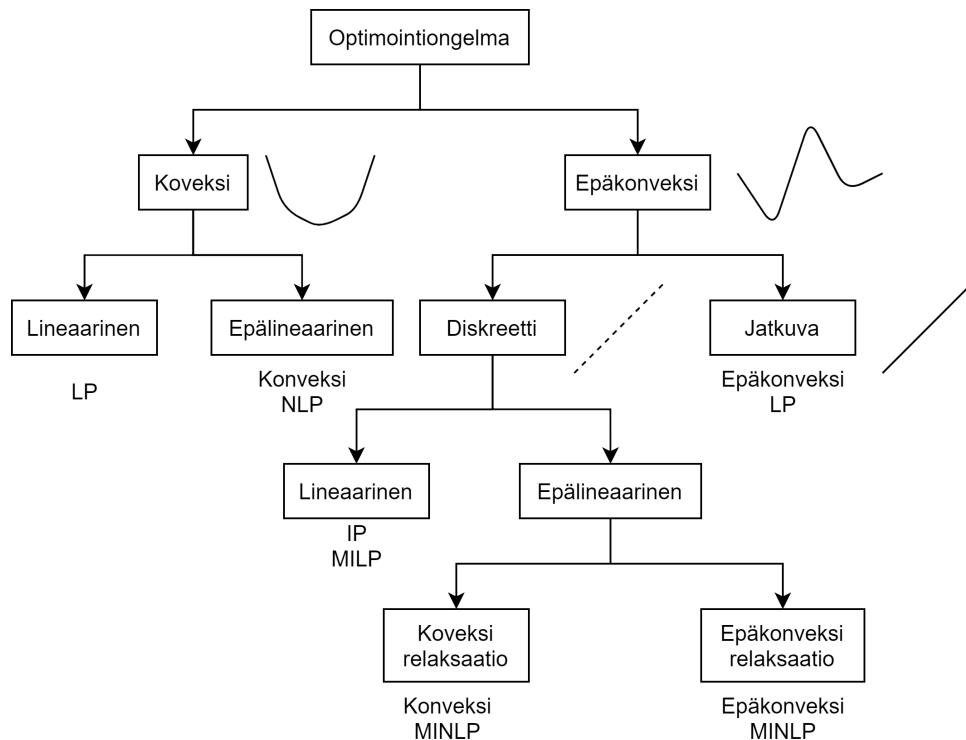
Optimoinilla tarkoitetaan systemaattista prosessia löytää paras ratkaisu tietyssä haku- ympäristössä, tavoitteena minimoida tai maksimoida haluttu funktio. Optimointiongelmat voidaan jaotella suunnittelumuuttujien tyypin ja tehtävän konveksisuuden perusteella. Kuvassa 2.1 on esitetty optimointiongelmien jaottelu Lin et al. [29] mukaan. Optimointia kutsutaan myös matemaattiseksi ohjelmoinniksi (mathematical programming). Matemaattinen optimointiongelma perusmuodossa:

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x}) \quad (2.1)$$

$$\text{siten että: } h_i(\mathbf{x}) = 0 \quad \forall i = 1, m \quad (2.2)$$

$$g_j(\mathbf{x}) \leq 0 \quad \forall j = 1, n \quad (2.3)$$

missä f on tehtävän kohdefunktio, \mathbf{x} on suunnittelumuuttujat sisältävä vektori, Ω on ha- kuvaruuden käypä joukko ja h_i ja g_j ovat yhtälö- sekä epäyhtälörajoitusehdot.



Kuva 2.1. Optimointiongelmien jaottelu Lin et. al mukaan [29]

Hakuavaruuden käypä joukko voidaan esittää matemaattisesti

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n \mid h_i(\mathbf{x}) = 0, g_j(\mathbf{x}) \leq 0, \forall i, j\} \quad (2.4)$$

Jos tehtävässä on sekä jatkuvia-, että diskeettejä muuttujia, voidaan muuttujavektori esittää kahtena erillisinä vektorina. Tyypillisesti vektorit ovat muotoa

$$\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \hat{\mathbf{Y}} \quad (2.5)$$

missä \mathbf{x} ja \mathbf{y} ovat jatkuvat- ja diskreetit muuttujat sisältävät vektorit ja $\hat{\mathbf{Y}}$ on diskreetit arvot sisältävä matriisi.

2.1 Rajoitettu optimointi

Optimointitehtävän tulosta voidaan rajoittaa rajoitusehtojen avulla. Rajoitusehtoja voidaan ajatella eräänlaisina leikkauspintoina, jotka leikkaavat hakuavaruudesta epäkäypä alueita pois. Suunnitteluavaruuden piste $\mathbf{x} \in \Omega$ on epäkäypä jos yksikin rajoitusehdoista rikkoutuu.

Rajoitusehdot ovat joko yhtälö- tai epäyhtälörajoitusehtoja, niiden perusmuodot on esitetty kaavoissa 2.2 ja 2.3. Rakenteiden optimoinnissa rajoitusehdot esitetään tyypillisesti normalisoidussa muodossa, jolloin rajoitusehdon arvo kuvaa suhteellista käyttöastetta [12]

$$g(\mathbf{x}) = \frac{E_d(\mathbf{x})}{R_d(\mathbf{x})} - 1 \leq 0 \quad (2.6)$$

missä E_d kuvaa sauvan rasitusta ja R_d -kestävyyttä.

Normalisoinnin tavoitteena on asettaa kaikki rajoitusehdot samalle asteikoille, jotta niiden keskinäinen vertaileminen olisi helpompaa. Rajoitusehdot voitaisiin esittää myös lineaarisessa muodossa

$$E_d(\mathbf{x}) - R_d(\mathbf{x}) \leq 0 \quad (2.7)$$

jolloin rajoitusehtojen arvot saattaisivat erota toisistaan monituhatkertaisesti. Tämä saattaa aiheuttaa optimointialgoritmile numeerisia haasteita, etenkin käytettäessä sakkofunktiota.

2.1.1 Sakkofunktio

Jotkin ratkaisualgoritmit, kuten geneettinen algoritmi, pystyvät ratkaisemaan vain rajoittamattomia tehtäviä. Rajoitettu tehtävä voidaan muuntaa rajoittamattomaksi ottamalla rajoitusehdot huomioon kohdefunktiossa sakkofunktion avulla. Sakkofunktio voi olla joko sisäinen, jolloin tehtävä pyritään pitämään käyvällä alueella tai ulkoinen, jolloin rikotut rajoitusehdot kasvattavat kohdefunktion arvoa. Käytettäessä ulkoista sakkofunktiota sakon tulee olla tarpeeksi suuri, jotta rajoitusehtoja rikotaan vain vähän. Suuret sakot saattavat kuitenkin aiheuttaa numeerisia vaikeuksia, joten hyvänä lähestymistapana pidetään aloittamista pienellä sakolla ja kasvattamalla sitä optimoinnin edetessä. [12]

Ulkaisen sakon menetelmässä vain rikotut rajoitusehdot otetaan huomioon sakkofunktiossa. Käytettäessä ulkoista sakkofunktiota, kohdefunktio on muotoa:

$$\min \Phi_k(\mathbf{x}) = f(\mathbf{x}) + \sigma_k \sum_j^n h_j(\mathbf{x})^2 + \sigma_k \sum_i^m c(g_i(\mathbf{x})) \quad (2.8)$$

$$c(g_i(\mathbf{x})) = \max(0, g_i(\mathbf{x}))^2 \quad (2.9)$$

missä σ_k on k :nnen iteraation sakkotermi.

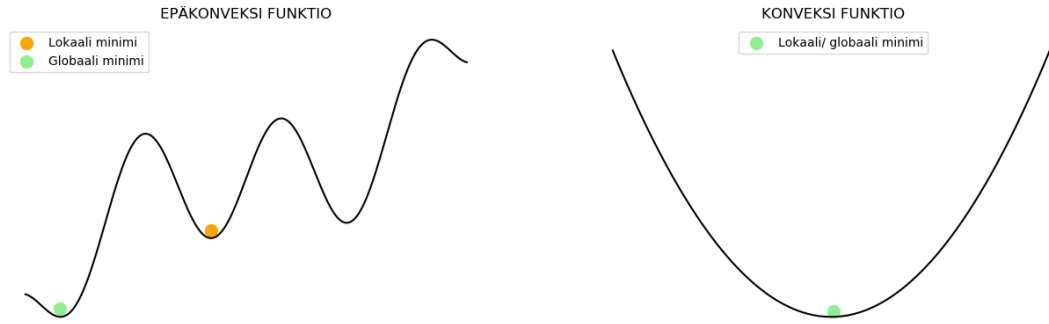
Sisäinen sakkofunktio, jota kutsutaan myös seinämämenetelmäksi (barrier method) toimii vain jos tehtävässä on pelkästään epäyhtälörajoituksia. Yksinkertaisin sisäinen sakkofunktio on käännetty seinämäfunktio (inverse barrier function), jossa jokaiselle rajoitusehdolle kirjoitetaan seinämäfunktio

$$B(\mathbf{x}) = -\frac{r_k}{g_i(\mathbf{x})} \quad (2.10)$$

Sisäistä sakkotermiä käytettäessä kohdefunktio on muotoa

$$\phi(\mathbf{x}) = f(\mathbf{x}) + r_k \sum_i^n B_i(\mathbf{x}) \quad (2.11)$$

Aloituspisteen on oltava käypä käytettäessä sisäistä sakkofunktiota. Seinämäfunktion arvo kerrotaan kertoimella r_k , jota pienennetään optimoinnin edetessä. Sisäisen sakon tarkoituksena on pyrkiä pitämään ratkaisu käyvän alueen sisällä. Suurella r_k :n arvolla sisäinen sakkofunktio luo suuren hypyn kohdefunktion arvolle lähestyttäessä rajoitusehdon rajaa, jolloin tämä seinämä pitää suunnittelumuuttujien arvot käyvällä alueella. [12] Käyvän aloituspisteen löytäminen saattaa olla hankalaa, joten sisäisen sakon menetelmä ei sovi tapauksiin, jossa aloituspiste valitaan sattumanvaraisesti.



Kuva 2.2. Epäkonveksissa funktiossa voi olla monta lokaalia minimiä

2.2 Konveksisuus

Joukko on konvekxi jos ja vain jos kaikkien pisteiden väliset janaat sisältyvät joukkoon. Funktio $f(\mathbf{x})$ on konvekxi jos kahdelle toisistaan eroavalle pisteelle \mathbf{x}_1 ja \mathbf{x}_2 pätee

$$f(a\mathbf{x}_1 + (1-a)\mathbf{x}_2) \leq af(\mathbf{x}_1) + (1-a)f(\mathbf{x}_2), \quad \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n \quad (2.12)$$

missä $0 \leq a \leq 1$, eli funktion arvot kahden pisteen $a\mathbf{x}_1$ ja $(1-a)\mathbf{x}_2$ välillä ovat aina pienempiä tai yhtäsuuria kuin kyseisten pisteiden välille muodostetun janan arvot.

Optimointitehtävä on konvekxi, jos rajoitusehtojen rajaama alue on konvekxi joukko ja kohdefunktio on konvekxi käyvässä joukossa. Esimerkki konveksista optimointitehtävästä on lineaarinen optimointitehtävä. Konveksilla optimointitehtävällä on olemassa ainoastaan yksi lokaali minimi tai maksimi, joka on samalla funktion globaali optimi. Kuvassa 2.2 on esitetty esimerkit konveksista ja epäkonveksista funktiosta sekä havainnollistettu globaalini- ja lokaalin minimin eroa.

Epäkonveksin optimointitehtävän ratkaisemisesta tekee haasteellisen globaalini optimin löytäminen, sillä lokaaleja optimeja saattaa olla monia eikä optimi ole rajattu ääripisteisiin. Eri aloituspisteet saattavat johtaa eri lopputuloksiin ja käyvän aloituspisteen löytäminen voi olla haastavaa.

2.3 Suunnittelumuuttujat

Suunnittelumuuttujilla tarkoitetaan optimoinnin muuttujia, joiden arvoa muuttamalla kohdefunktio pyritään minimoimaan tai maksimoimaan. Ne saavat joko diskreettejä tai jatkuvia arvoja. Diskreetit arvot voivat olla esimerkiksi kokonaislukuja tai arvoja ennalta määritetystä listasta. Suunnittelumuuttujien valinnalla on suuri rooli ratkaisun löytämisen kannalta, sillä tehtävän kohdefunktio ja rajoitusehdot muodostetaan valittujen suunnittelumuuttujien avulla.

Indeksimuuttujat ovat erikoistapauksia kokonaislukumuuttujista. Niiden arvo viittaa niille määritetyn diskreetin joukon järjestysnumeroon. Indeksimuuttujaa voidaan käyttää esimerkiksi rakenneosan profiilin muuttujana, jossa muuttujan joukkona on lista eri profiilivaihtoehtoista. Kuvassa 2.3 on havainnollistettu indeksimuuttujien toimintaperiaatetta.

Indeksi	0	1	2	3	4	5
	IPE 100	IPE 120	IPE 140	IPE 160	IPE 180	IPE 200

Kuva 2.3. Indeksimuuttujan saadessa arvon 3, valikoituu profiiliksi IPE 160

Diskreetit arvot voidaan esittää myös binäärimuuttujien ja jatkuvan muuttujan avulla. Diskreetit arvot relaksoidaan ja niiden arvoa kuvataan jatkuvalla muuttujalla X ja jokaista diskreettiä arvoa \hat{X}_i kohti luodaan binäärimuuttuja y_i , jonka avulla jatkuvan muuttujan arvo pakotetaan diskreetiksi. Binäärimuuttujat voivat saada arvon 0 tai 1. Jatkuvan muuttujan arvon muodostumista on havainnollistettu kuvassa 2.4. Käytettäessä tätä menettelyä, on tehtävälle lisättävä kaksi uutta lineaarista rajoitusehtoa

$$X = \sum_i^k \hat{X}_i y_i \quad (2.13)$$

$$\sum_i^k y_i = 1 \quad (2.14)$$

missä \hat{X}_i on diskreetit arvot sisältävä lista.

$$\begin{array}{c}
 \mathbf{y} \\
 \hline
 0 \\
 1 \\
 0 \\
 0 \\
 1
 \end{array}
 *
 \begin{array}{c}
 \hat{\mathbf{X}} \\
 \hline
 100 \\
 150 \\
 200 \\
 250 \\
 300
 \end{array}
 = 150 + 300 = 450$$

Kuva 2.4. Jatkuvan muuttujan arvon muodostuminen binäärimuuttujien avulla. Muuttujan arvo ei ole kuitenkaan käypä, sillä binäärimuuttujien summa on 2

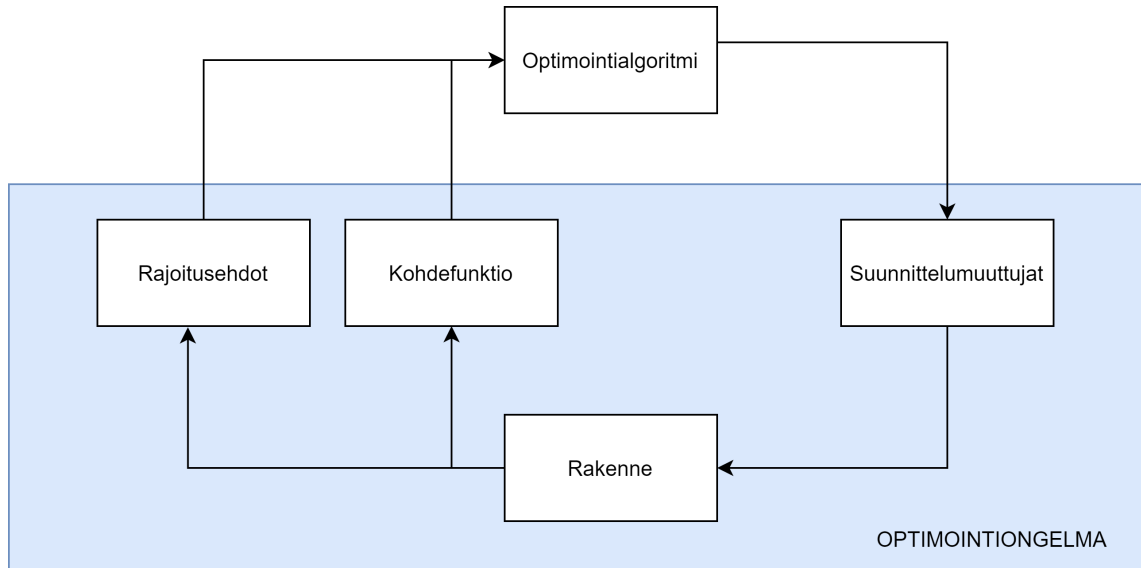
2.4 Optimoinnin toteutus

Ongelman muotoilu ja määrittäminen ovat tärkeitä vaiheita rakenteiden optimoinnissa. Optimointimenetelmät saattavat johtaa absurdeihin ja rakenteiden mekaniikan teoriaa rikkoviin ratkaisuihin, jos tehtävä ei ole tarpeeksi tarkkaan rajattu. [1] Monissa tapauksissa osa rakenneosien välisistä suhteista on implisiittisiä eli epäsuoria, joten analyttisiä riippuvuussuhteita ei pystytä kirjoittamaan, vaan tulos saadaan erilaisten simulaatioiden, kuten elementtimenetelmän (Finite Element Method, FEM) avulla.

Tässä työssä optimointiongelmat ratkaistaan Python -ohjelmointikielellä kirjoitetulla ohjelmalla, joka optimoi sekä analysoi rakenteen. Optimointitehtävä muodostuu optimoitavasta rakenteesta, suunnittelumuuttujista, kohdefunktiosta sekä rajoitusehdoista. Suunnittelumuuttujille määrätään kohde, kuten poikkileikkaus tai materiaali, mihin muuttuja vaikuttaa ja ominaisuus, kuten poikkileikkauksen leveys tai materiaalin lujuus, jonka arvoa muutetaan. Samalla muuttujalla voi olla useita kohteita ja ominaisuuksia, mutta vain yksi arvo ja yhdet muuttujarajat. Esimerkiksi neliöputkiprofiileista muodostetun ristikon tapauksessa yhdellä suunnittelumuuttujalla voidaan muuttaa kaikkien uumasauvojen korkeutta ja leveyttä.

Jokaiselle rajoitusehdolle määritetään tarvitseeko sen evaluoiminen FEM-analyysiä. Jos laskenta on jo suoritettu, eikä yksikään suunnittelumuuttuja ole muuttunut, rakenteen analysointia ei suoriteta turhaan uudestaan. Kun suunnittelumuuttujien arvot muuttuvat, suoritetaan FEM-laskenta rajoitusehtojen evaluoinnin yhteydessä. Näin pyritään minimoimaan FEM-analyysien määrä ja varmistumaan siitä, että rajoitusehtojen arvot vastaavat aina nykyistä tilannetta.

Optimointialgoritmi muuttaa suunnittelumuuttujien arvoja, jonka seurauksena optimoitava rakenne muuttuu. Tämän jälkeen uusi rakenne analysoidaan ja rajoitusehtojen sekä kohdefunktion arvo palautetaan optimointialgoritmillemme. Näiden arvojen avulla algoritmi tekee muutoksia suunnittelumuuttujien arvoihin. Iterointia jatketaan kunnes määritetty lopetusehto täyttyy. Optimoinnin kulkua on havainnollistettu kuvassa 2.5.



Kuva 2.5. Optimoinnin kulku

3 DISKREETIN OPTIMOINNIN MENETELMIÄ

Teräsrakenteiden diskreettiä optimointia on tutkittu jo yli 50 vuotta. Raymond Toakley esitti vuoden 1968 julkaisussaan [33], että on mahdollista johtaa menetelmiä, jotka eivät riipu teoreettisista paino-kestävyys funktiosta, vaan käyttävät suoraan käytössä olevia diskreettejä poikkileikkauksia.

Optimointitehtävän rajoitusehtoina käytetään paikallisen standardin rajoja, esimerkiksi Eurokoodin [6] mukaisia rajoja, joita ovat muunmuassa erilaiset jännitys- ja siirtymäraajat. Diskreetin optimoinnin kannalta on olennaista rajoittaa suunnittelumuuttujien arvot profiilikatalogin mukaisiin arvoihin.

Käytettäessä standardoituja poikkileikkauksia, tehtävä on eräänlainen erikoistapaus diskreettien muuttujien ongelmasta, koska poikkileikkauksen ominaisuudet eivät ole täysin toisistaan riippumattomia. Gradienttipohjaiset menetelmät eivät välttämättä sovellu tämän tyyppisiin tehtäviin, jossa poikkileikkauksen ominaisuuksien suhdetta ei pysty esittämään analyttisesti. [18]

Diskreetin optimoinnin menetelmät voidaan jakaa karkeasti neljään eri kategoriaan [18]:

- Yhden muuttujan menetelmät, jossa muut ominaisuudet approksimoidaan
- Yhden muuttujan menetelmät ilman approksimointia
- Usean muuttujan menetelmät
- Pyöristysmenetelmät

Usein optimitulosta on kuitenkin vaikea löytää vain yhtä menetelmää käyttämällä. Monessa tapauksessa onkin hyödyllistä yhdistellä näitä menetelmiä ja ratkaista tehtävä monessa vaiheessa yhdistelemällä eri menetelmien hyötyjä.

3.1 Yhden muuttujan approksiomenetelmät

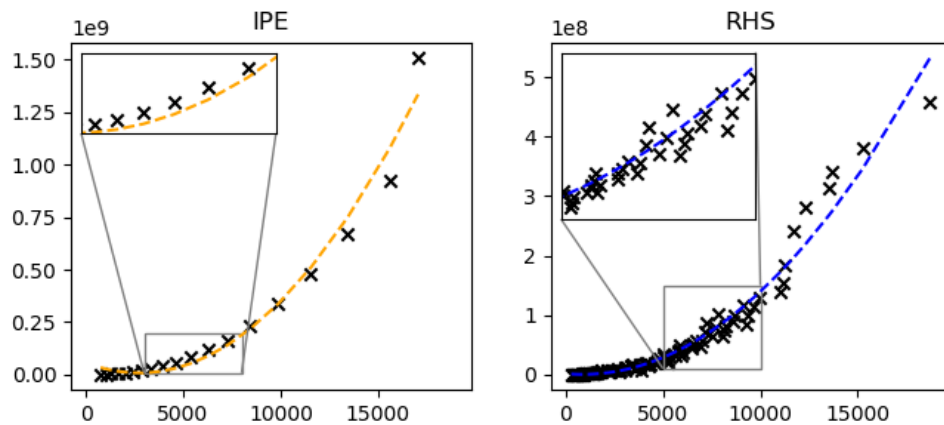
Tässä menetelmätyypissä yhtä poikkileikkauksen ominaisuutta käsitellään jatkuvana muuttujana ja muut profiilin ominaisuudet lausutaan tämän muuttujan suhteen. Koska muuttujat ovat jatkuvia, voidaan gradienttipohjaisia menetelmiä hyödyntää ongelman ratkaisemisessa. Usein muuttujien väliset riippuvuussuhteet eivät kuitenkaan ole lineaarisia, jolloin muuttujien välistä suhdetta tulee kuvata tarkemmin erillisellä sovitefunktiolla. Esimerkiksi

poikkileikkauksen neliömometti voidaan approksimoida pinta-alan suhteen

$$I_i = \gamma_I(A_i) \quad (3.1)$$

missä γ_I on funktio, joka kuvaa pinta-alan ja neliömomentin välistä suhdetta.

Kaikissa ongelmissa ei ole kuitenkaan mahdollista käyttää vain yhtä ominaisuutta ainoana muuttujana, sillä muita ominaisuuksia ei pysty tarkasti määrittämään vain yhden muuttujan avulla. I-profiilien osalta tämän tyyppinen approksimointi yhden muuttujan avulla on mahdollista, mutta muilla profiilityypeillä ominaisuuksien arvoja ei välttämättä pystytä esittämään riittävän tarkasti yhden muuttujan avulla. Kuvassa 3.1 on esitetty neliömometti y-akselin ympäri pinta-alan suhteen. Kuva havainnollistaa kuinka IPE-poikkileikkausten ominaisuuksille saa muodostettua tarkan sovitteen, mutta suorakaidepoikkileikkauksille ei. Lisäksi I/H-profiileilla on eri perheitä kuten IPE, HEA ja HEB, eikä näille saada muodostettua yhteistä tarkkaa sovitetta, vaan jokaiselle perheelle on muodostettava omansa.



Kuva 3.1. RHS -profiileille on vaikea löytää sovitetta jolla saisi tarkasti approksimoitua poikkileikkauksen ominaisuuksia

Menetelmää voidaan hyödyntää monivaiheessa optimoinnissa hakuavaruuden rajaamiseen. Optimointi on nopeaa, sillä suunnittelumuuttujia on vähän ja gradienttipohjaisia menetelmiä pystytään hyödyntämään. Menetelmän heikkoutena voidaan pitää sen joustamattomuutta. Käytettävät profiilityytit on ennalta määrättävä, sillä jokaisella poikkileikkaustyytillä on omanlaisensa sovitefunktiensa ja käytettävissä olevan listan muuttuessa sovitefunktiot on johdettava uudelleen. [18]

3.2 Yhden muuttujan menetelmät ilman approksimaatiota

Sen sijaan, että käyttää approksimoituja ominaisuuksien välisiä suhteita, voidaan yhtä muuttujaa, esimerkiksi pinta-alaa, käyttää yksinään ja laskennan aikana muut profiilin ominaisuudet luetaan taulukosta. Ongelmasta tulee yleisempi, koska kaikki profiilin ominaisuudet ovat riippuvaisia vain yhdestä muuttujasta. Näiden ominaisuuksien suhde ei

kuitenkaan ole jatkuva tai derivoituva, joten gradienttipohjaisia menetelmiä ei voida hyödyntää. [18]

Toinen lähestymistapa on käyttää indeksimuuttujia eli profiilin järjestysnumeroja suunnittelumuuttujina. Kun profiilin indeksi on päätetty, voidaan kaikki sen ominaisuudet helposti lukea taulukosta ja käyttää kaikissa laskuissa. Riippuen käytettävästä ratkaisumenetelmästä, tulee käytettävien profiilien taulukko järjestää valitun ominaisuuden mukaan nousevassa järjestyksessä. Tapa on suosittu erityisesti käytettäessä metaheuristisia optimointialgoritmeja [17] [27] [16].

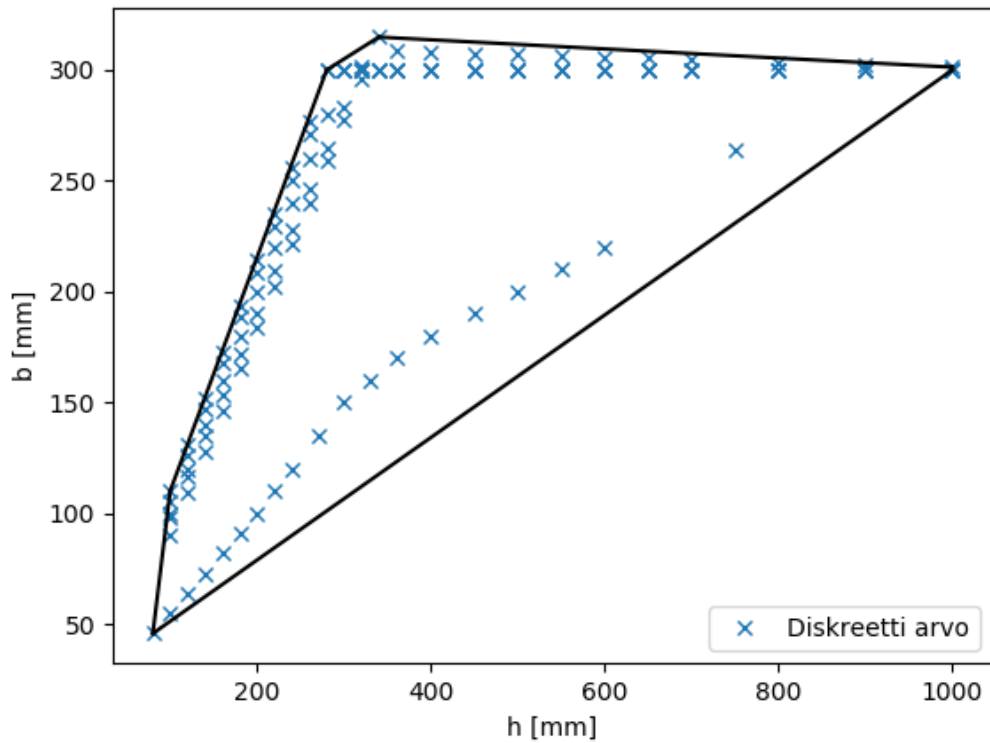
3.3 Usean muuttujan menetelmät

Tässä menetelmässä käytetään osaa poikkileikkauksen ominaisuuksista riippumattomina suunnittelumuuttujina. Hager ja Balling [13] kehittivät kaksivaihetekniikan, jossa suunnittelumuuttujina käytettiin pinta-alaa, taivutusvastusta sekä neliömomenttia. Ensimmäisessä vaiheessa tehtävä relaksoitiin ja ratkaistiin jatkuvien muuttujien avulla. Ongelmaksi muodostui se, että optimaalinen ratkaisu ei ollut lähelläkään diskreettiä ratkaisua, koska arvojen suhteet ovat sellaisia joita ei löydy profiilikatalogista. Yksi ratkaisu tähän ongelmaan on muodostaa koveksi verho profiilien ominaisuuksien avulla, joka luo uusia lineaarisia rajoitusehtoja ongelmalle pakottaen lopputuloksen olemaan lähellä diskreettiä ratkaisua. [13]

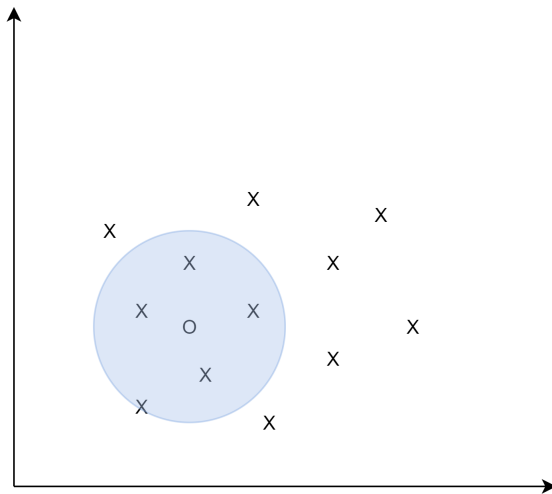
Toisessa vaiheessa diskreetti ratkaisu pyrittiin löytämään muunnetulla Branch and Bound -menetelmällä. Profiilien läpikäynnin nopeuttamiseksi määritettiin pieni kolmen tai neljän poikkileikkauksen ympäristö, josta ratkaisua lähdettiin etsimään. Vaikka haku ympäristöä pienennettiin, tuli kokeellisten ratkaisujen määrä silti suureksi, vaatien suurta laskennallista tehoa rakenteiden analyysiin.

Käytettäessä profiilin dimensioita suunnittelumuuttujina voidaan kaikki tarvittavat ominaisuudet laskea optimoinnin aikana. Huonona puolena voidaan pitää sitä, että poikkileikkaustyyppiä ei voi vaihtaa optimoinnin aikana, esimerkiksi pilarin profiili pysyy I-profiilina vaikka optimaalisin poikkileikkaus saattaisikin olla suorakaideputkiprofiili.

Pinta-alaa, taivutusvastusta sekä muita poikkileikkauksen ominaisuuksia käytettäessä, profiilityyppi ratkeaa käytettävissä olevien profiilien ja jatkuvan optimituloksen mukaan. Osaa standardin mukaisista rajoitusehdoista, kuten poikkileikkausluokitusta ja liitoksen kestävyyttä, ei kuitenkaan pystytä ratkaisemaan optimoinnin aikana. Tämä saattaa johtaa tilanteeseen, missä jatkuvan ratkaisun ympärillä olevat lähimmät diskreetit profiilit eivät täytäkään tehtävän todellisia rajoitusehtoja.



Kuva 3.2. I-poikkileikkauksien korkeudelle ja leveydelle muodostettu konvekssi verho



Kuva 3.3. Toisessa vaiheessa käytettävät diskreetit arvot voidaan valita jatkuvan ratkaisun ympäriltä esimerkiksi euklidisen etäisyyden perusteella

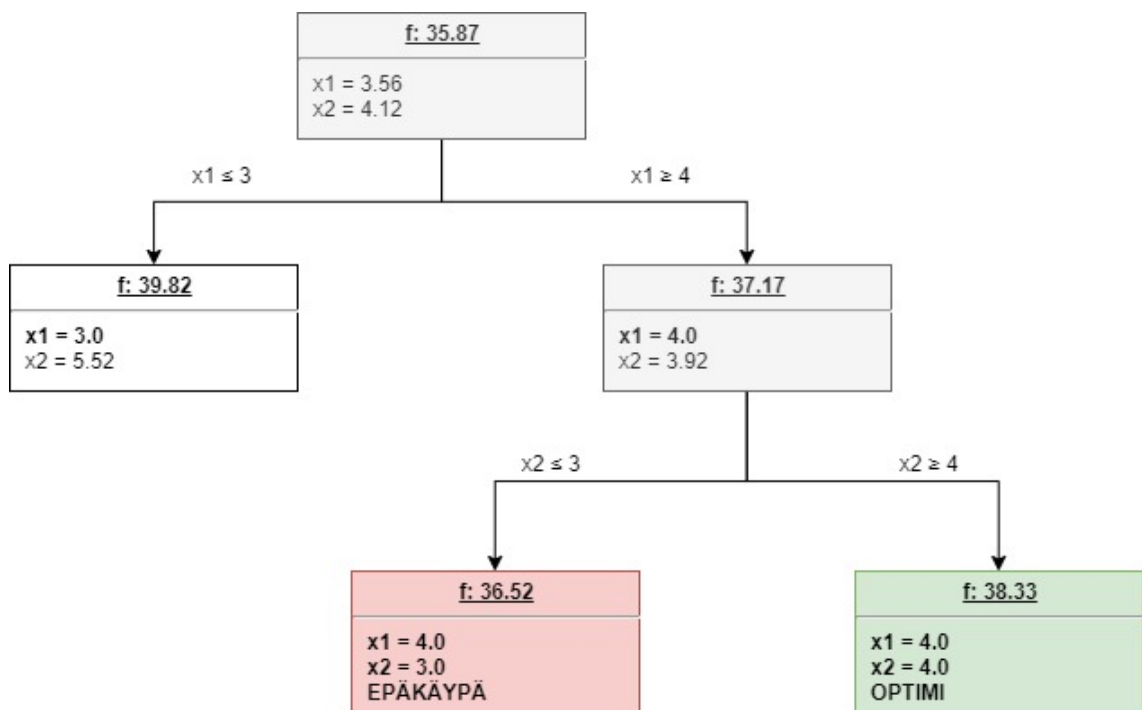
3.4 Pyöristysmenetelmät

Pyöristysmenetelmillä tarkoitetaan jatkuvan ratkaisun diskretointia. Jos ongelma voidaan esittää ja ratkaista jatkuvien muuttujien avulla, niin helpoin ja yksinkertaisin menetelmä diskreetin ratkaisun saavuttamiseksi on pyöristää jatkuva muuttuja ylöspäin lähimpään diskreettiin arvoon. Tämä menetelmä voi kuitenkin johtaa rakenteen ylipainoon tai epä-

käypään diskreettiin ratkaisuun [35].

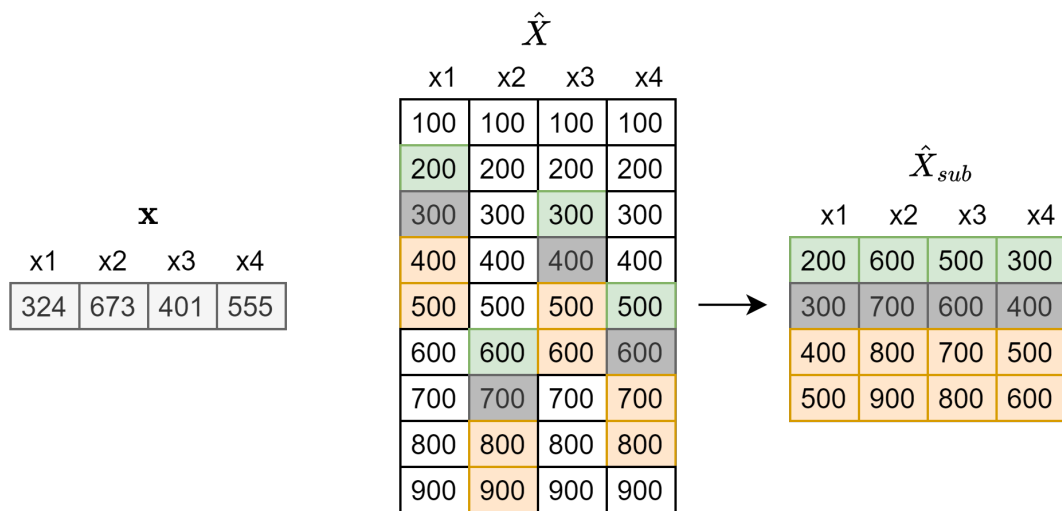
Branch and Bound (BnB) -menetelmä on luultavasti tunnetuin ja useimmin käytetty menetelmä diskreetin ratkaisun löytämiseksi [35]. Jatkuvan tehtävän ratkaisua voidaan pitää alarajaratkaisuna, jonka ympäristöstä diskreetti optimi pyritään löytämään. Menetelmässä pyritään löytämään diskreetti ratkaisu yksi muuttuja kerrallaan haaroittamalla tehtävä kahdeksi alitehtäväksi, jossa muuttujan arvot rajoitetaan lähimpään suurempaan ja pienempään diskreettiin arvoon uusien rajoitusehtojen avulla. Kun muuttujalle on löydetty diskreetti arvo, siirrytään seuraavaan muuttujaan ja haaroittamista jatketaan kunnes kaikille halutuille muuttujille on löydetty diskreetti arvo.

Menetelmä perustuu peräkkäisiin analyysihin diskreetin hakupuun jokaiselle muuttujalle, jonka seurauksena laskenta-aika kasvaa eksponentiaalisesti suunnittelumuuttujien määrän kasvaessa. Tämän seurauksena menetelmä ei sovellu globaaliksi hakumenetelmäksi elementtimenetelmällä analysoitaviin tehtäviin jossa on monta muuttujaa. Vaikka BnB löytää varmasti globaalin optimin lineaarisessa ja konveksissa tehtävissä, menetelmä ei kuitenkaan välttämättä suppene yleisissä epäkonvekseissa ongelmissa. [35]



Kuva 3.4. Branch and Bound -menetelmän toimintaperiaate

Toinen suosittu tapa on valita jatkuvan ratkaisun lähin diskreetti arvo ja tämän ympäriltä k alemmaa ja l ylempää diskreettiä arvoa. Diskreettien arvojen etäisyys jatkuviin arvoihin voidaan määrittää esimerkiksi Euklidisella etäisyydellä. Tästä $n(k + l + 1)$, missä n on suunnittelumuuttujien lukumäärä, kokoisesta kandidaattijoukosta diskreetti optimiratkaisu pyritään löytämään käyttämällä esimerkiksi luvussa 3.2 esitettyjä indeksimuuttujia sekä sopivaa optimointialgoritmia. Kandidaattijoukon muodostamista on havainnollistettu kuvassa 3.5.



Kuva 3.5. Kandidaattijoukon muodostaminen jatkuvan ratkaisun lähimmän diskreetin arvon ympäriltä

4 TUTKIMUKSESSA KÄYTETTÄVÄT ALGORITMIT

Valittaessa sopivaa algoritmia optimointitehtävän ratkaisemiseksi on tunnettava optimointiongelma sekä erilaiset algoritmit hyvin. Ei ole olemassa yhtä algoritmia, jolla voitaisiin ratkaista kaikki optimointiongelmat. Tähtävän luonteesta ja suunnittelumuuttujista riippuen voidaan käyttää joko deterministisiä tai metaheuristisia algoritmeja.

4.1 Deterministiset algoritmit

Deterministiset algoritmit ovat numeerisia menetelmiä, jotka pyrkivät löytämään optimituloksen tarjoten samalla teoreettisen todistuksen siitä että löydetty ratkaisu on oikeasti optimi. Lähdetäessä samasta alkupisteestä ja käytettäessä samoja parametreja, deterministiset algoritmit päätyvät aina samaan lopputulokseen.

4.1.1 Peräkkäisten linearisointien menetelmä, SLP

Peräkkäisten linearisointien menetelmä (Sequential Linear Programming, SLP) on iteraatiivinen optimointimenetelmä, jossa optimipiste pyritään löytämään ratkaisemalla iteraatiivisesti tehtävän linearisoitu aliongelma. Menetelmä soveltuu jatkuvan ja derivoituvan tehtävän ratkaisemiseen. Tässä työssä linearisoitu alitehtävä ratkaistaan Googlen OR-tools:iin implementoidulla COIN-OR:n branch and cut -algoritmeilla [8].

Epälineaarille fuktioille voidaan kirjoittaa lineaarinen approksimaatio Taylorin sarjan ensimmäisen termin avulla

$$f(\mathbf{x}_{k+1}) \approx f(\mathbf{x}_k) + \nabla f^T(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k) \quad (4.1)$$

missä \mathbf{x}_k on nykyinen iteraatiopiste ja funktion gradientti $\nabla f(\mathbf{x}_k)$ voidaan määrittää joko analyttisesti tai numeerisesti.

Jos hakusuunta \mathbf{d} on määritetty, sievenee kaava muotoon:

$$f(\mathbf{x}_{k+1}) \approx f(\mathbf{x}_k) + \nabla f^T(\mathbf{x}_k)\mathbf{d} \quad (4.2)$$

Epäkonvekseissa ongelmassa osa linearisoinneista saattaa leikata pois osan käyvästä

alueesta, jolloin globaalien ongelmien ratkaisu voi jäädä linearisoidun käyvän alueen ulkopuolelle [24]. Lineaarinen approksimaatio ei kuvaa tarkasti epälineaarisia rajoitusehtoja, joten vaikka alitehtävän ratkaisu olisi käypä, se ei välttämättä ole pätehtävässä käypä. Koska linearisoitu funktio kuvaa epälineaarista funktiota tarkasti vain linearisointipisteen ympärillä, tulee linearisoidun alitehtävän muuttujat rajoitetaan välille

$$\mathbf{x}_{L_k} \leq \mathbf{x}_k \leq \mathbf{x}_{U_k} \quad (4.3)$$

missä \mathbf{x}_{L_k} ja \mathbf{x}_{U_k} muuttujavektorin uudet siirtymäraajat.

Tämän työn SLP:n implementoinnissa siirtymäraajat muodostetaan jokaiselle alitehtävälle seuraavasti

$$\begin{aligned} \mathbf{x}_{L_k} &= \max(\mathbf{x}_k - m_L \Delta, \mathbf{x}_L) \\ \mathbf{x}_{U_k} &= \min(\mathbf{x}_k + m_U \Delta, \mathbf{x}_U) \\ \Delta &= \mathbf{x}_U - \mathbf{x}_L \end{aligned} \quad (4.4)$$

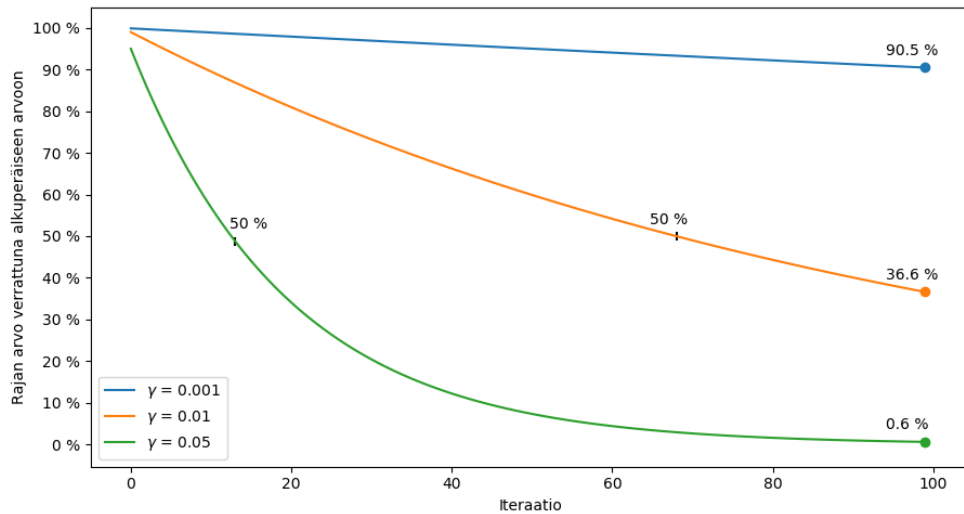
missä m_L ja m_U ovat ala- ja ylärajan positiiviset muutoskerroimet, \mathbf{x}_L ja \mathbf{x}_U ovat suunnittelumuuttujien alkuperäiset muuttujarajat.

Siirtymärajoja tulee myös tiivistää iteroinnin edetessä, koska muuten optimitulos saattaa jäädä vaihtelevaan kahden tai useamman pisteen välille. Siirtymärajojen huolellinen valinta on tärkeää ratkaisun löytämisen kannalta ja se vaatii SLP -menetelmän tuntemista sekä tietoa ratkaistavasta ongelmasta. Arora ehdottaakin kirjassaan [1] käyttäjiä kokeilemaan rohkeasti erilaisia rajoja, jos yksi määrittäminen johtaa epäkäypään tulokseen. Muutoskerroimia pienennetään pienennyskerroimella γ jokaisella iteraatiokierroksella seuraavasti:

$$m_{k+1} = (1 - \gamma)m_k \quad (4.5)$$

Kuvassa 4.1 on esitetty pienennyskerroimen vaikutus siirtymärajoihin sadan iteraation aikana. Mitä suurempi kerroin, sitä nopeammin hakualaue suppenee, mikä voi johtaa tulokseen kaukana optimista. Toisaalta, liian pienellä pienennyskerroimen arvolla algoritmi saattaa jäädä oskilloimaan usean pisteen välille pitkäksi aikaa, jolloin optimoinnin kesto kasvaa huomattavasti.

Approksimaatiosta johtuva epätarkkuus ja siirtymäraajat saattavat yhdessä johtaa tilanteeseen missä linearisoidulla alitehtävällä ei ole ratkaisua nykyisessä rajoitetussa ympäristössä. Tilanteessa, missä aloituspiste on epäkäypä normalisoitujen rajoitusehtojen suhteen ja siirtymäraajat ovat tiukat, siirtymärajojen muodostama alue voi pysyä kokonaan epäkäyvällä alueella johtaen epäkäypään ratkaisuun. Tässä tilanteessa on hyödyllistä relaxoida rajoitusehdot ensimmäisten iteraatiokierrosten ajaksi. Relaxointi voidaan to-



Kuva 4.1. Pienennyskerroimen γ :n vaikutus muuttujarajojen arvoon iteroinnin edetessä

teuttaa lisäämällä tehtävälle uusi muuttuja β [12]. Relaksoitu alitehtävä on nyt muotoa

$$\begin{aligned}
 \min f(\mathbf{x}) + C\beta \\
 g_i(\mathbf{x}) - \beta \leq 0 \\
 h_j(\mathbf{x}) - \beta = 0 \\
 \beta \geq 0, \quad \beta \in \mathbb{R}^+
 \end{aligned} \tag{4.6}$$

missä C on suuri kerroin, jotta optimointikierrokset pyrkivät pienentämään β :n arvoa enemmän suhteessa $f(\mathbf{x})$:n arvoon.

Linearisoidun alitehtävän muodostaminen saattaa viedä suurimman osan optimointiin kuluva ajasta. Monimutkaisissa rakenteissa FEM-analyysin tekeminen on raskasta ja usein suunnittelumuuttujia on monia, jonka seurauksena alitehtävän muodostamiseen kuluva aikaa kasvaa eksponentiaalisesti tehtävän koon kasvaessa.

4.1.2 Peräkkäinen sekalukutehtävän linearisointi, MISLP

Sekalukuoptimoinnilla (Mixed-Integer Programming, MIP) tarkoitetaan optimointiongelmää, jossa suunnittelumuuttujina käytetään sekä jatkuvia että diskreettejä muuttujia. Peräkkäinen sekalukutehtävän linearisointi (Mixed-Integer Sequential Linear Programming, MISLP) menetelmä toimii samalla periaatteella kuin SLP, mutta nyt aliongelmana on linearisoitu sekalukutehtävä jatkuvan tehtävän sijasta. Linearisoitu sekalukutehtävä ratkaistaan Googlen OR-tools:iin implementoidulla COIN-OR:n branch and cut -ratkaisijalla [8].

Tämän työn MISLP implementoinnissa hyödynnetään diskreettimuuttujia, joille annetaan vektori käytössä olevista diskreeteistä arvoista. Ongelma linearisoidaan jatkuvien muuttujien avulla ja tehtävälle muodostetaan alitehtäväksi lineaarinen sekalukutehtävä, jonka ratkaisuna on seuraava iteraatiopiste. Jokaista siirtymärajojen sisällä olevaa diskreettiä arvoa vastaan luodaan binäärimuuttuja, joka pakottaa ratkaisun diskreetiksi. Alitehtävässä on siis kahdentyyppisiä muuttujia, jatkuvia- sekä binääri- eli kokonaislukumuuttujia, mikä tekee tehtävästä sekalukutehtävän.

Tarkastellaan binäärimuuttujien käyttöä esimerkin avulla. Kootaan käytössä olevat pinta-alat vektoriksi $\mathbf{A} = [100, 200, 300]$ ja käytetään jatkuvalla pinta-alamuuttujalle merkintää \hat{A} . Alitehtävässä jokaista listan alkioita vastaan lisätään alitehtävälle uusi binäärimuuttuja. Binäärivektori on siis muotoa $\mathbf{b} = [b_1, b_2, b_3]$. Tehtävälle lisätään myös kaksi uutta lineaarista rajoitusehtoa, joiden avulla ratkaisu pakotetaan diskreetiksi

$$\begin{aligned} \hat{A} &= \mathbf{A}^T \mathbf{b} \\ \sum_{i=1}^3 b_i &= 1 \end{aligned} \tag{4.7}$$

missä yksittäinen binäärimuuttuja b_i , voi saada joko arvon 0 tai 1.

Lisätyt rajoitusehdot yhdessä pakottavat jatkuvan muuttujan saamaan diskreetin arvon. Binäärivektorin summan ollessa 1 vain yhden binäärimuuttujan arvo on 1, jonka seurauksena jatkuvan muuttujan arvo on oltava yksi muuttujalle alunperin määritetyistä diskreeteistä arvoista.

$$\begin{aligned} \mathbf{b} &= [0, 0, 1] \\ \hat{A} &= \mathbf{A}^T \mathbf{b} = [100, 200, 300]^T \cdot [0, 0, 1] = 300 \end{aligned} \tag{4.8}$$

Algoritmin muuttujarajat ja pienennyskerroin määritetään samalla tavalla kuin SLP:lle. Tässä työssä MISLP:n muuttujarajoina käytetään aina vähintään yhtä suurempaa ja pienempää diskreettiä arvoa, jotta ei päädyttäisi tilanteeseen missä diskreettiä ratkaisua ei voida löytää. Tämän seurauksena epäkonvekseissa ongelmissa algoritmi saattaa jäädä oskilloimaan kahden tai useamman pisteen välille.

4.2 Metaheuristiset algoritmit

Metaheuristiset algoritmit kuvaavat yleensä jotain luonnonilmiötä, kuten parviälyä ja ne ovat nykyään käytetyimpien optimointialgoritmien joukossa. Niillä on monta etua tavallisiin deterministisiin algoritmeihin verrattuna, kuten kyky käsitellä epäjatkuvia ongelmia. [25]

Monet metaheuristiset menetelmät ovat stokastisia eli satunnaisuudella on suuri vaikutus

lopputulokseen. Satunnaisuus mahdollistaa hakuavaruuden laajemman tutkimisen, joka suosii erilaisten ratkaisujen löytymistä. [9] Stokastisuuden takia kertaalleen löydettyä optimitulosta on vaikea toistaa tulevilla optimointikerroilla.

4.2.1 Muuttuvan ympäristön haku, VNS

Muuttuvan ympäristön haku (Variable Neighbourhood Search, VNS) on joustava kehys luoda heuristiikkoja kombinatoristen ja epälineaaristen optimointiongelmiin likimääräiseen ratkaisemiseen. [15] Menetelmän toiminta perustuu hakuavaruuden jakamiseen pienempiin lokaaleihin hakuympäristöihin ja niiden iteratiiviseen ratkaisemiseen.

VNS pyrkii systemaattisella hakuympäristön muuttamisella löytämään optimi- tai likimääräisen optimituloksen seuraavien huomioiden pohjalta [15] :

1. Lokaali optimi yhdessä ympäristössä ei välttämättä ole lokaali optimi toisessa ympäristössä
2. Globaali optimi on lokaali optimi kaikissa ympäristöissä
3. Empiirinen tutkimus osoittaa että kaikki tai suurin osa lokaaleista optimeista on suhteellisen lähellä toisiaan. [23]

Yksinkertaisin VNS-algoritmi koostuu hakuympäristön määrittämisestä, satunnaisen aloituspisteen valinnasta, paremman pisteen löytämisestä ja hakuympäristön muuttamisesta. Hakuympäristö $\mathcal{N}(\mathbf{x})$ määritetään annetun mittafunktion δ suhteen nykyisen pisteen \mathbf{x} ympärillä seuraavasti [15]:

$$\mathcal{N}_k(\mathbf{x}) = \{\mathbf{y} \in \mathbf{X} \mid \delta(\mathbf{x}, \mathbf{y}) \leq \alpha\}, \quad \alpha > 0 \quad (4.9)$$

missä \mathbf{X} on koko tehtävän hakuavaruus, \mathbf{x} on uuden hakuympäristön pisteet ja α positiivinen luku.

Satunnaisen aloituspisteen valinnalla on tarkoitus välttää lokaalin minimin lähiympäristöön juuttuminen. Määritetään $\mathcal{N} = \{\mathcal{N}_1 \dots \mathcal{N}_{k_{max}}\}$, jossa jokainen operaattori \mathcal{N}_k , $1 \leq k \leq k_{max}$ määrittää uuden hakuympäristön $\mathcal{N}_k(\mathbf{x})$ nykyisen pisteen \mathbf{x} ympärillä. Satunnainen piste \mathbf{x}' valitaan hakuympäristöstä seuraavasti

$$\mathbf{x}' = \mathbf{R}(\mathcal{N}_k(\mathbf{x})) \quad (4.10)$$

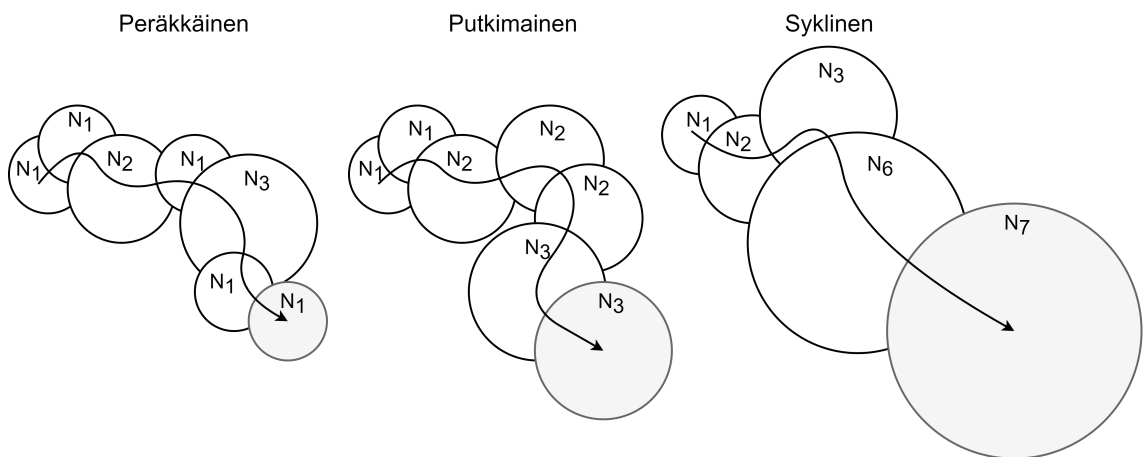
missä \mathbf{R} on satunnaisoperaattori.

Uusi hakuympäristö voidaan määrittää peräkkäisenä-, syklisenä-, putkimaaisena- tai vino- na hakuympäristön muuttamisena. Kuvassa 4.2 on visualisoitu uusien hakuympäristöjen muodostumista. Hakuympäristön muuttamisella tarkoitetaan hakuoperaattorin \mathcal{N}_k muuttamista, jonka avulla nykyisen pisteen \mathbf{x} hakuympäristö määritetään. Peräkkäinen haku- ympäristön muuttaminen (sequential neighbourhood change) palauttaa hakuympäristön

aina pienimpään \mathcal{N}_1 ympäristöön kun parempi piste on löydetty. Jos parempaa pistettä ei löydetä, kasvatetaan nykyistä haku ympäristöä.

Syklisessä haku ympäristön muuttamisessa (cyclic neighborhood change) haku ympäristöä kasvatetaan joka iteraatiokierroksella vaikka parempi tulos löydetäisiin nykyisestä ympäristöstä.

Putkimainen haku ympäristön muutos (pipe neighborhood change) kasvattaa haku ympäristöä, jos parannusta nykyiseen ei löydy nykyisessä ympäristössä. Erona peräkkäiseen muutokseen haku ympäristön kokoa ei palauteta alkuperäiseen, vaan etsintää jatketaan samankokoisessa ympäristössä.



Kuva 4.2. Haku ympäristön muutostavat visualisoituna

Vinossa haku ympäristön muutoksessa (skewed neighborhood change) voidaan hyväksyä parempien pisteiden lisäksi myös huonompia pisteitä, jos ne ovat tarpeeksi kaukana nykyisestä pisteestä. Tämän tarkoituksena on tutkia lokaaleja minimejä kaukaa nykyisestä pisteestä. [15] Vinoa haku ympäristön muutosta käytettäessä kohdefunktion voidaan ajatella olevan muotoa

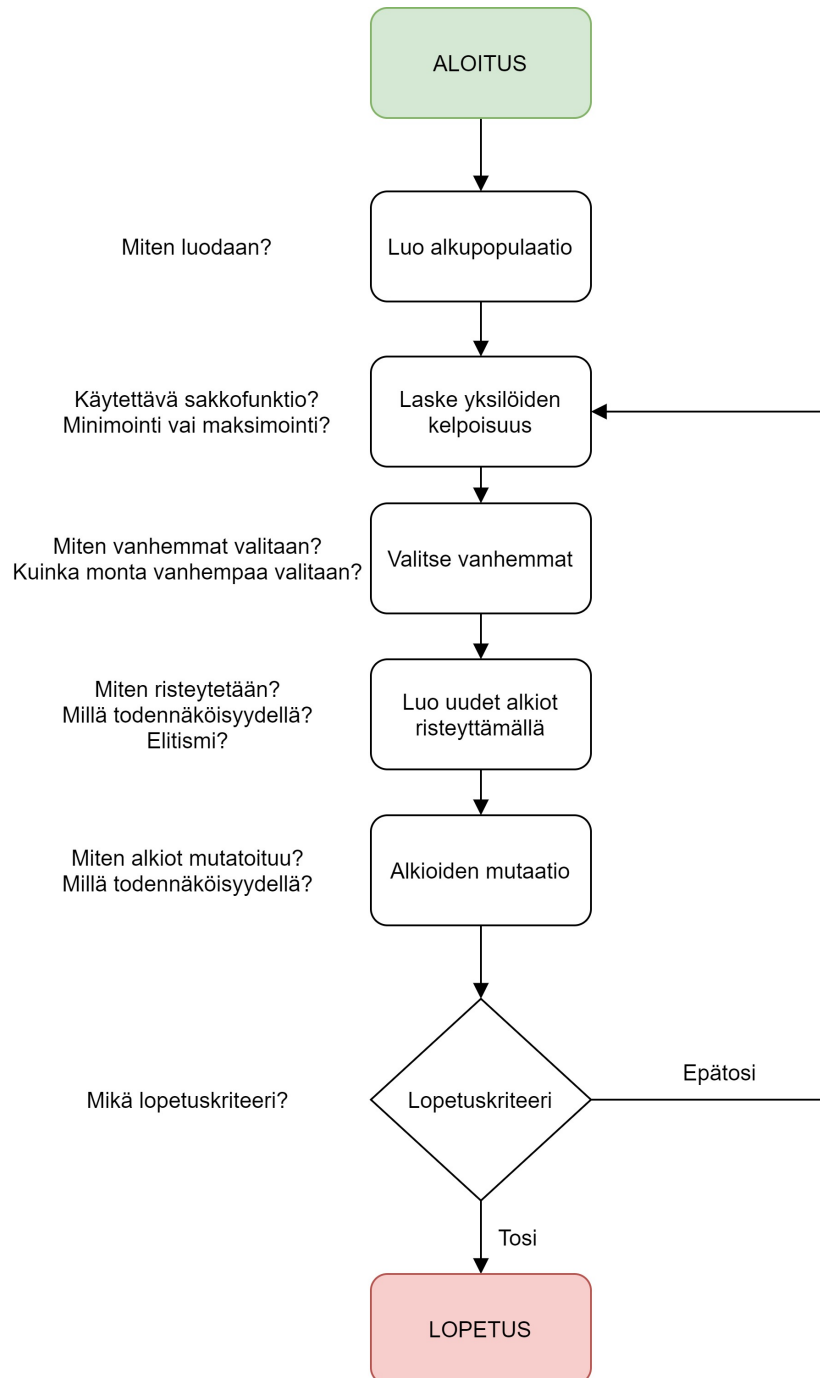
$$\min g(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) - f(\mathbf{y}) - \alpha \delta(\mathbf{x}, \mathbf{y}) \quad (4.11)$$

missä α on positiivinen kerroin ja $\delta(\mathbf{x}, \mathbf{y})$ on pisteiden \mathbf{x} ja \mathbf{y} välinen etäisyys.

Haku ympäristön tutkimiseen voidaan käyttää monia erilaisia metaheuristisia menetelmiä kuten adaptiivista monialoitusta (adaptive multi-start), jäähydysmenetelmää (Simulated Annealing, SA) tai geneettisiä algoritmeja. [14] Ensimmäisen parannuksen strategiassa uuteen pisteeseen siirrytään heti kun parempi piste on löytynyt, kun taas parhaan parannuksen strategiassa haetaan optimi nykyisestä hakuavaruudesta ja muodostetaan uusi haku ympäristö tämän pisteen suhteen.

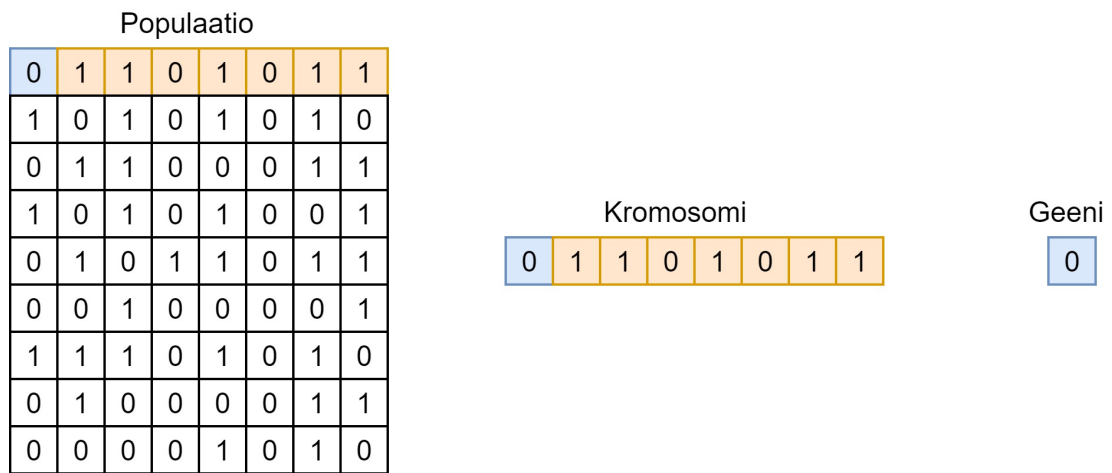
4.2.2 Geneettinen algoritmi, GA

Geneettiset algoritmit (genetic algorithm, GA) ovat algoritmeja jotka matkivat luonnon evoluutiota. Ne ovat luonteeltaan heuristisia sekä stokastisia ja soveltuvat rajoittamattoman optimointitehtävän ratkaisemiseen. Geneettiset algoritmit ovat suosittuja tehtävissä, missä hakuavaruus on suuri ja jossa likimääräinenkin optimi riittää ratkaisuksi. Geneettisen algoritmin toiminta ja eri vaiheissa pohdittavia kysymyksiä on esitetty vuokaaviona kuvassa 4.3.



Kuva 4.3. Geneettisen algoritmin vuokaavio ja eri vaiheissa pohdittavia kysymyksiä

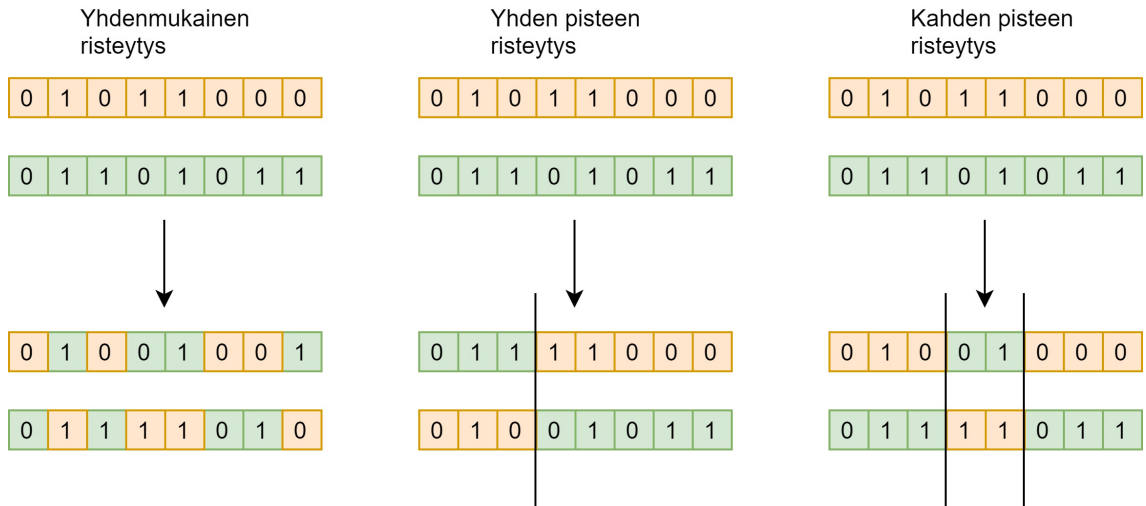
Amerikkalaista John Hollandia pidetään nykyaikaisten geneettisten algoritmien kehittäjänä. Hän oli ensimmäinen joka käytti geneettisiä operaatioita, kuten risteytystä ja mutaatioita adaptiivisten ja keinotekoisien menetelmien tutkimisessa. [25] Geneettisillä algoritmeilla on kaksi pääetua verrattuna tavallisiin algoritmeihin: mahdollisuus käsitellä komplekseja ongelmia sekä rinnakkaisuus. Olipa kohdefunktio konvekssi, epäkonvekssi, lineaarinen tai epälineaarinen, voidaan sitä käsitellä genettisten algoritmien avulla. Populaatiopohjainen lähestymistapa mahdollistaa rinnakkaisen implementaation, jossa jokainen populaation yksilö voidaan käsitellä yhtäaikaaisesti, joka nopeuttaa algoritmin toimintaa. [25] Kromosomi vastaa muuttujavektoria, eli siitä erotellaan eri muuttujien binäärilukuketjut. Kuvassa 4.5 on havainnollistettu populaation muodostumista.



Kuva 4.4. Populaatio muodostuu kromosomeista, jotka muodostuvat geneistä

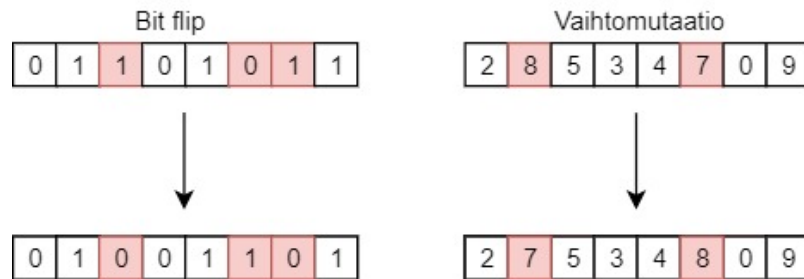
Kolme pääkomponenttia geneettisissä algoritmeissa ovat risteytys, mutaatio ja parhaan yksilön valinta. Jokainen ratkaisu koodataan listaksi lukuja, yleensä binääri- tai kokonaislukuja, jota kutsutaan kromosomiksi. Kahden vanhemman risteytyksessä uudet yksilöt muodostetaan vaihtamalla osa vanhempien kromosomien geneistä. Uuden yksilön muodostamiseen risteytyksellä on usein suuri todennäköisyys, tyypillisesti 80 - 95 % [25]. Risteytyksessä yksilö perii osan vanhempien geneistä valitun menetelmän mukaisesti. Risteytysmenetelmiä ovat esimerkiksi yhden tai useamman pisteen risteytys (single- / k-point crossover) sekä yhdenmukainen risteytys (uniform crossover). Yhden pisteen risteytyksessä valitaan satunnainen kohta kromosomista jonka suhteen vanhempien geenit vaihdetaan. Yhdenmukaisessa risteytyksessä toisen vanhemman geenit käydään läpi yksi kerrallaan ja vaihdetaan toisen vanhemman vastaavaan geeniin todennäköisyydellä p . Erilaisia risteytysmenetelmiä ja uusien alkuiden muodostumista on havainnollistettu kuvassa 4.5.

Mutaatiossa yksilön kromosomin genejä muutetaan valitun mutaatiomenetelmän mukaisesti. Näitä ovat esimerkiksi binäärimuotoisen kromosomin bitin kääntö -mutaatio (flip bit) tai vaihtomutaatio (swap mutation). Bitin kääntö -mutaatiossa sattumanvaraisesti valitut geenit muutetaan nollassa ykköseksi tai päinvastoin, kun taas vaihtomutaatiossa sattumanvaraisesti valitut geenit vaihtavat keskenään paikkaa. Yksilön ja geenin mutaatiotodennäköisyys on yleensä alhainen 0.1 - 5 % [25]. Edellä mainitut mutaatiomenetelmät on



Kuva 4.5. Risteytysmenetelmiä havainnollistettuna

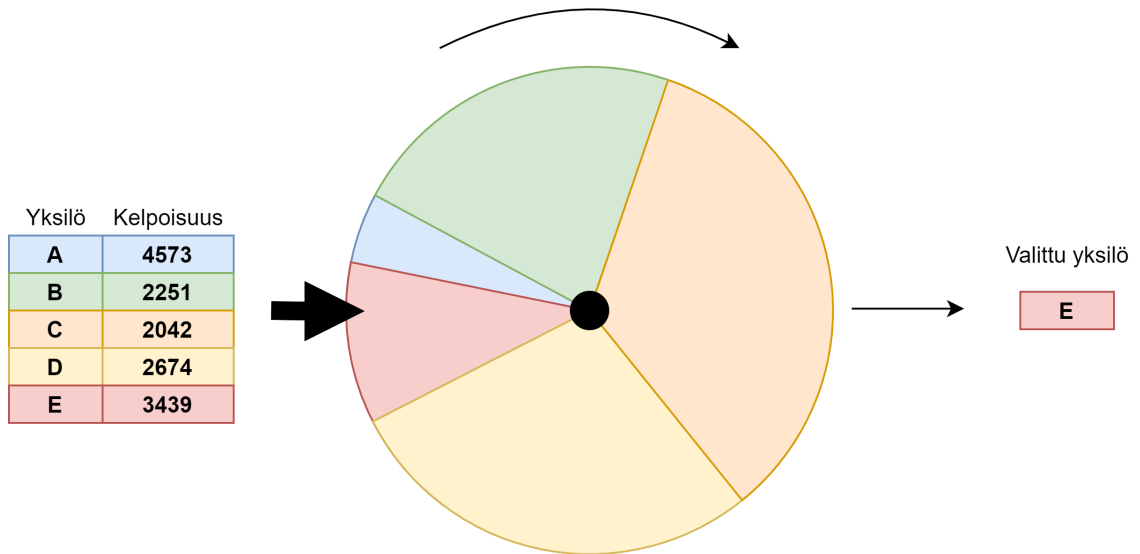
havainnollistettu kuvassa 4.6.



Kuva 4.6. Mutaatio vaikuttaa kromosomin yksittäisiin geeneihin

Uuden populaation luomisen jälkeen uusien yksilöiden kelpoisuus lasketaan ja parhaat yksilöt valitaan muodostamaan joukko vanhempia. Vanhempia valitaan n -kappaletta, kuitenkin vähintään kaksi. Valittaessa useampi vanhempi, muodostetaan risteytettävät vanhemmat satunnaisesti kaikista vanhemmista. Jos nykyinen paras yksilö halutaan säilyttää uudessa joukossa, voidaan se lisätä uuteen populaatioon ilman muutoksia. Tätä menettelyä kutsutaan elitismiksi. Vanhempien valinta voidaan suorittaa valitsemalla suoraan n parasta yksilöä populaatiosta. Muita menetelmiä vanhempien valitsemiseksi on esimerkiksi turnausmenetelmä (tournament selection), jossa valitaan satunnaisesti k yksilöä populaatiosta joista pätevin valitaan uudeksi vanhemmaksi tai rulettimenettely (roulette wheel selection), jossa jokainen yksilö saa kelpoisuuttaan vastaavan suuruisen osan rulettipyörästä jota pyörytetään ja kohdalle osunut yksilö valitaan vanhemmaksi. Rulettimenettelyn toimintaperiaate on esitetty kuvassa 4.7.

Geneettisillä algoritmeilla lopetuskriteereinä pidetään tyypillisesti generaatioiden tai kohdefunktion evaluointien ylärajan saavuttamista [31]. Järkevien rajojen asettaminen algoritmille vaatii tuntemusta ratkaistavasta ongelmasta, jotta hakuavaruus ei paisu tarpeettoman suureksi. Algoritmin suorittaminen voidaan myös lopettaa, jos todennäköisyys merkittäviin muutoksiin seuraavien generaatioiden aikana on pieni. Käytettäessä tätä kolmatta, adaptiivista vaihtoehtoa, tietoa ratkaistavasta ongelmasta ei vaadita.



Kuva 4.7. Rulettimenetelmässä vanhemmat valitaan simuloitun rulettipyöräytyksen tuloksena

Todennäköisyys merkittäviin muutoksiin voidaan määrittää joko genotyyppisesti (genotypical) tai fenotyyppisesti (phenotypical) [31]. Genotyyppisesti määrätyssä lopetuskriteerissä algoritmin suorittaminen lopetetaan, kun käyttäjän asettama prosenttiosuus geenejä populaatiosta on konvergoitunut, esimerkiksi 80 %. Geenin voidaan sanoa olevan konvergoitunut, kun suurimmalla osalla populaatiosta on samalla geenillä sama arvo. Fenotyyppisessä lopetuskriteerissä tarkastellaan algoritmin n viimeisimmän generaation edistymistä, missä n on käyttäjän määrittämä luku. Kun tarkkailtava arvo, kuten populaation keskimääräinen kohdefunktion muutos, johtaa arvoon alle ennalta asetetun rajan ϵ , algoritmin voidaan sanoa olevan konvergoitunut ja iterointi lopetetaan välittömästi. [31]

Geneettiset algoritmit ovat suosittuja rakenteiden optimoinnissa. Niitä on käytetty esimerkiksi luvun 5 rakenteiden, teräksisen kattoristikon [4] sekä ontelolaattalattioiden optimoinnissa [2]. Niiden suosio rakenteiden optimoinnissa pohjautuu niiden kykyyn käsitellä epä-jatkuvia ja/tai epäderivoituvia ongelmia, iterointi voidaan aloittaa epäkäyvästä pisteestä ja lopputulokseksi riittää usein hyvä, käypä ratkaisu.

5 DISKREETIN OPTIMOINNIN VERROKKITEHTÄVIÄ

Tässä luvussa esitetään kirjallisuudessa yleisesti käytettyjä rakenteita eri optimointimenetelmien vertailemiseen. Tulos hyväksytään, jos se rikkoo rajoitusehtoja enintään 0.0001.

SLP ja MISLP-algoritmeille käytetään seuraavia parametrien arvoja

$$\begin{aligned}
 m_L = m_U &= 0.1 \\
 \gamma &= 0.01 \\
 \beta &= 10 \\
 C &= 200000
 \end{aligned}
 \tag{5.1}$$

VNS:n lokaalina hakualgoritmina käytetään genettistä algoritmia ja populaation kokona $n = 100$. Kahden pisteen risteytys tapahtuu 80 % todennäköisyydellä, yksilö mutatoituu 5 % todennäköisyydellä ja mutatoituessaan yksittäisen geenin arvo muuttuu nykyisessä hakuympäristössä satunnaisesti 5 % todennäköisyydellä. Populaation vanhemmiksi valitaan kaksi populaation pätevintä yksilöä.

Tehtävissä käytetyt diskreetit poikkileikkauskatalogit on esitetty liitteessä A.

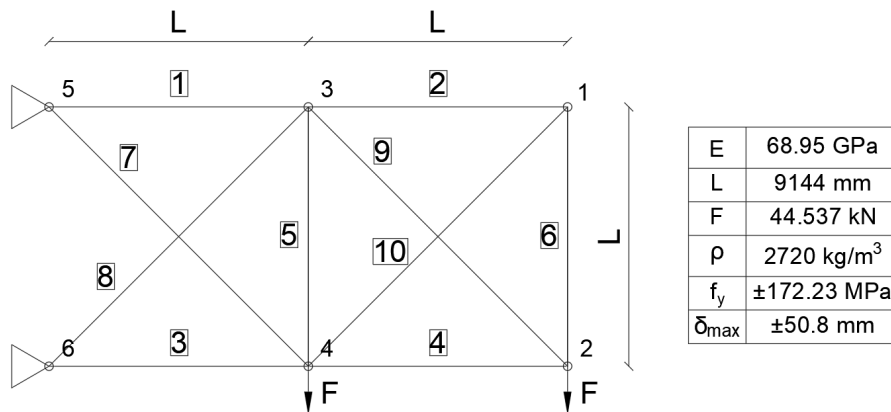
5.1 10-sauvan ristikko

10-sauvan ristikko on yksi yleisimmistä rakenteista, jonka avulla eri algoritmeja vertailaan. Tehtävän tavoitteena on minimoida kuvan 5.1 mukaisen ristikkorakenteen massa. Suunnittelumuuttujina käytetään diskreettejä sauvojen pinta-aloja ja rajoitusehtoina ovat sauvojen normaalijännitykset sekä rakenteen solmupisteiden siirtymät. Sauvojen pinta-alat valitaan taulukon A.1 mukaisista pinta-aloista. Mahdollisia ratkaisuja on 42^{10} eli yli 17 tuhatta biljoonaa, joten tehtävän ratkaiseminen kaikki vaihtoehdot läpikäymällä ei ole mahdollista.

Tehtävä matemaattisessa muodossa:

$$\begin{aligned}
 \min \rho \sum_{i=1}^{10} L_i x_i \\
 \frac{|N_i(\mathbf{x})|}{f_y A_i} - 1 \leq 0 \\
 \frac{|\delta_i(\mathbf{x})|}{\delta_{max}} - 1 \leq 0 \\
 x_i \in \mathbf{A}_{10} \\
 \mathbf{x} = [A_1, A_2, \dots, A_{10}]
 \end{aligned} \tag{5.2}$$

Juang ja Chang [20] ratkaisivat 10-sauvan ristikon käyttämällä diskreettiä Lagrangen funktion metodia (Discrete Lagrangian Method, DLM). Menetelmän toimivuutta testattiin lähtemällä liikkeelle viidestä eri aloituspisteestä: alarajalta, ylärajalta, keskiarvosta, aiemmin löydetystä optimituloksesta [3] sekä satunnaisesta aloituspisteestä. Heidän menetelmällään päädyttiin optimitulokseen 2489.95 kg jokaisessa tapauksessa lähtöpisteestä huolimatta.



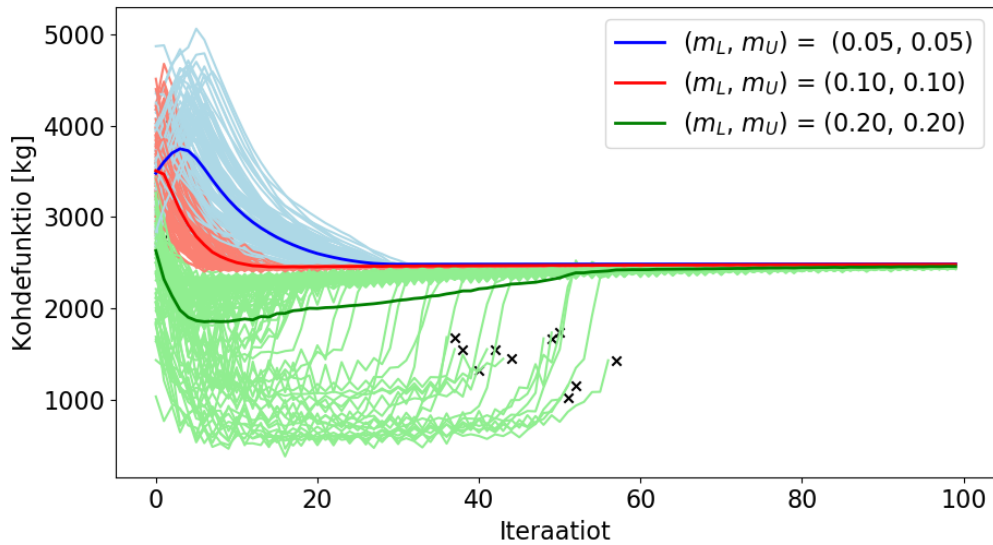
Kuva 5.1. 10-sauvan ristikko ja tehtävässä käytettyjen parametrien arvot

Tässä työssä 10-sauvan ristikko ratkaistaan kaksivaihe tekniikalla. Ensimmäisessä vaiheessa tehtävä relaxoidaan ja ratkaistaan SLP:llä jatkuvien muuttujien avulla ja toisessa vaiheessa ratkaisu diskretoidaan VNS:n avulla, jolloin käytössä on diskreetit muuttujat. Lähtöpisteeksi valitaan kokeilemalla sauvoille pienin pinta-ala, jolla tehtävä on vielä käypä. Lähtöpisteeksi valikoituu $\mathbf{x}_0 = [12835.5, 12835.5, \dots, 12835.5]$. Diskreetti lähtöpiste muodostetaan valitsemalla jatkuvan ratkaisun poikkileikkauksille lähimmät diskreetit arvot euklidisen etäisyyden avulla

$$\hat{\mathbf{x}}_i = \arg \min(\mathbf{A}_{10} - \mathbf{x}_i)^2 \tag{5.3}$$

missä $\hat{\mathbf{x}}_i$ on sauvan i diskreetti aloitusarvo.

Kuvassa 5.2 on esitetty miten SLP:n iterointi etenee satunnaisesta lähtöpisteestä eri muuttujarajoja käytettäessä. Vain 25 % tapauksissa muuttujarajan arvoilla (0.2, 0.2) al-



Kuva 5.2. Optimitulokseen päästään SLP-algoritilla lähes joka kerta satunnaisesta aloituspisteestä. Vain suurilla muuttujarajan arvoilla algoritmi saattaa jäädä jumiin epäkäypään lokaaliin minimiin.

goritmi jää jumiin epäkäypään lokaaliin minimiin. Kuvasta huomataan myös, että pienillä muuttujarajojen arvoilla tehtävä pyrkii käypään pisteeseen mahdollisimman nopeasti.

Taulukko 5.1. 10-sauvan ristikon optimitulokset

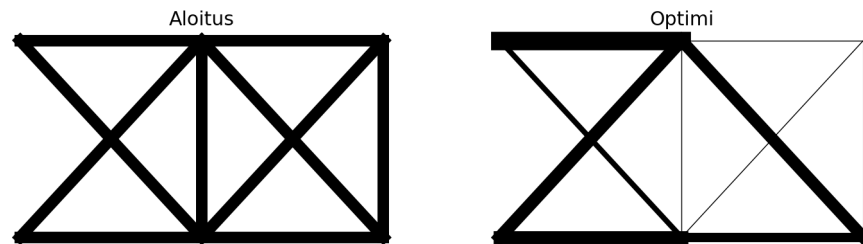
Muuttuja	GA [10]	PSO [27]	DLM [20]	SLP	SLP + VNS
A_1	21607.5	19350.0	21607.5	20771.91	21607.5
A_2	1044.9	1044.9	1044.9	1044.9	1044.9
A_3	14770.5	14770.5	14770.5	14977.08	14770.5
A_4	9159.0	8707.5	9159.0	9809.49	9159.0
A_5	1044.9	1044.9	1044.9	1044.9	1044.9
A_6	1044.9	1044.9	1044.9	1044.9	1044.9
A_7	5140.65	5140.65	5140.65	5404.22	5140.65
A_8	14770.5	17092.5	14770.5	14688.99	14770.5
A_9	14190.0	14190.0	14190.0	13910.76	14190.0
A_{10}	1044.9	1161.0	1044.9	1044.9	1044.9
Analyyseiden lkm	-	-	-	3900	6130 ± 560
Paino (kg)	2489.95	2508.65	2489.95	2487.01*	2489.95

* Tulos jatkuvien muuttujien avulla

Taulukossa 5.1 on esitetty optimitulokset kootusti. Diskreetti ratkaisu on hyvin lähellä jatkuvan tehtävän ratkaisua. Kaksivaiheinen menetelmä perustuukin tähän oletukseen, että tehtävän diskreetin ratkaisu on löydettävissä jatkuvan tehtävän ratkaisun lähistöltä. VNS löytää diskreetin ratkaisun jatkuvan tehtävän ympäriltä päätyen samaan ratkaisuun jul-

kaisujen [10] ja [20] kanssa.

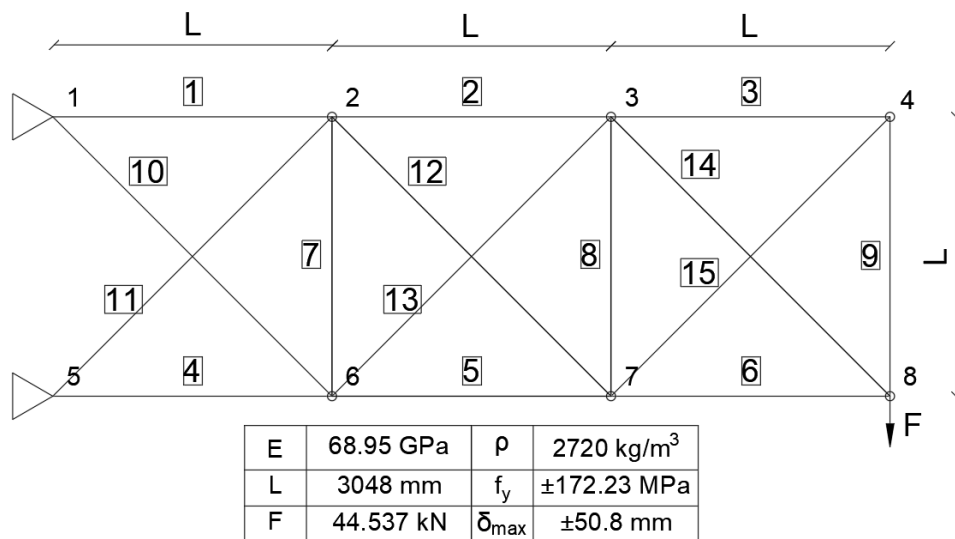
Kriittiseksi rajoitusehdoksi muodostuu solmun 2 siirtymä. Kuvasta 5.3 nähdään, että ristikon topologia ei ole optimaalinen, vaan siinä on turhia sauvoja.



Kuva 5.3. 10-sauvan ristikon optimiratkaisu

5.2 15-sauvan ristikko

Tässä tehtävässä tavoitteena on optimoida kuvan 5.4 mukaisen ristikkorakenteen geometria sekä sauvojen poikkileikkaukset. Rajoitusehtoina ovat sauvojen veto- ja puristusjännitykset. Kohdefunktiona käytetään rakenteen massaa, diskreetteinä suunnittelumuuttujina sauvojen poikkileikkausten pinta-aloja ja jatkuvina muuttujina solmujen koordinaatteja. Käytössä olevat pinta-alat on esitetty liitteen A taulukossa A.2. Koska tehtävässä on sekä jatkuvia, että diskreettejä muuttujia on mahdollisia vaihtoehtoja ääretön määrä.



Kuva 5.4. 15-sauvan ristikko ja tehtävässä käytettyjen parametrien arvot

Tehtävä matemaattisessa muodossa

$$\begin{aligned} \min \rho \sum_{i=1}^{15} L_i(\mathbf{x}, \mathbf{y}) A_i \\ \frac{|N_i(\mathbf{A}, \mathbf{x}, \mathbf{y})|}{f_y A_i} - 1 \leq 0 \\ A_i \in \mathbf{A}_{15} \end{aligned} \quad (5.4)$$

Geometriamuuttujien rajat on esitetty taulukossa 5.2. Tässä työssä geometriamuuttujien arvot on sidottu globaaliin koordinaatistoon, mutta arvot voitaisiin sitoa myös solmujen alkuperäisiin sijainteihin, jolloin kaikille muuttujille voitaisiin käyttää samoja muuttujarajoja.

Taulukko 5.2. 15-sauvan ristikon geometriamuuttujien rajat

Muuttuja	Alaraja	Yläraja
x_2	254	3556
x_3	5588	6604
x_6	254	3556
x_7	5588	6604
y_3	2540	3556
y_4	1270	2286
y_6	-508	508
y_7	-508	508
y_8	508	1524

Shojaee et. al [32] tutkivat parannellun diskreetin parveilualgoritmin (Improved Discrete Particle Swarm Optimization, IDPSO) ja liikkuvien asymptoottien menetelmän (Method of Moving Asymptotes, MMA) yhdistämistä tehtävän ratkaisemiseksi. PSO:n stokastisen luonteen vuoksi tehtävä optimoitiin 10 kertaa ja paras tulos ilmoitettiin.

Tässä työssä ratkaistaan relaxoitu tehtävä jatkuvien muuttujien avulla SLP-algoritmeilla sekä diskreetti tehtävä hyödyntäen MISLP-algoritmin binääriformulointia. Tavoitteena on vertailla jatkuvan ja diskreetin ratkaisun eroa. Ongelmaksi SLP- ja MISLP -algoritmeille muodostui FEM-analyysin tekeminen tilanteessa, missä solmut 4 ja 8 sulautuivat, jolloin solmujen välisen elementin pituus on nolla. Tilanteeseen voitiin joutua joko tehtävän linearisoinnissa tai alitehtävän ratkaisussa. Tehtävälle asetettiin uusi lineaarinen rajoitusehto, jotta solmut eivät voisi olla päällekkäin.

$$h(\mathbf{y}) : y_4 - y_8 \leq \epsilon \quad (5.5)$$

missä ϵ on jokin pieni luku, tässä työssä $\epsilon = 0.001$.

Aloituspisteenä käytetään kuvan 5.4 mukaista rakennetta ja kaikille sauvoille kokeilemalla löydettyä pineintä pinta-alaa, jolla tehtävä on käypä $A_i = 697.418, \quad \forall i \in [1, 15]$.

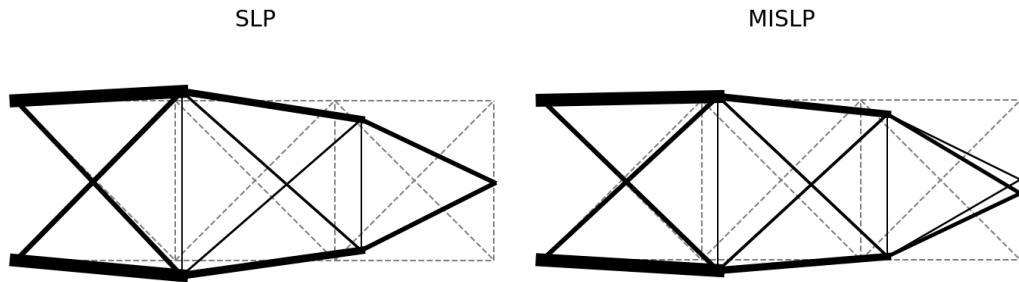
Tulokset on esitetty taulukossa 5.3. Erot alkuperäisissä julkaisuissa olevien kokonaispainojen ja taulukossa esitettyjen arvojen välillä johtuvat muunnoksesta brittiläisen yksikköjärjestelmän ja SI-järjestelmän mukaisten arvojen välillä. Taulukkoon on laskettu rakenteiden massat lähteissä [30] ja [32] esitettyjen tulosten avulla.

Taulukko 5.3. 15-sauvan ristikon optimitulokset

Muuttuja	IDPSO-MMA [32]	GA [30]	SLP	MISLP
A ₁	757.4	757.4	615.64	615.5
A ₂	347.7	347.7	345.54	347.7
A ₃	91.0	185.2	217.27	91.0
A ₄	615.5	615.5	601.68	615.5
A ₅	347.7	347.7	363.08	347.7
A ₆	185.2	91.0	221.23	185.2
A ₇	71.6	71.6	71.6	71.6
A ₈	71.6	71.6	71.6	71.6
A ₉	91.0	347.7	126.54	91.0
A ₁₀	223.9	283.9	232.7	223.9
A ₁₁	283.9	347.7	250.61	223.9
A ₁₂	185.2	174.2	129.49	141.9
A ₁₃	185.2	141.9	104.43	141.9
A ₁₄	185.2	91.0	71.6	185.2
A ₁₅	91.0	185.2	71.6	91.0
x ₂	2540	2580.07	3170.6	3363.9
x ₃	5833.3	5788.94	6604.0	6603.9
y ₂	3326.35	3423.88	3222.38	3117.91
y ₃	3244.89	3256.8	2687.45	2781.82
y ₄	1335.16	1393.52	1487.26	1534.36
y ₆	-321.77	-417.79	-291.48	-201.06
y ₇	-508	-337.84	189.38	66.0
y ₈	1131.16	1393.37	1467.22	1271.6
Analyysien lkm	-	-	14760	13542
Paino (kg)	33.56	34.18	31.77*	32.06

* Tulos jatkuvien muuttujien avulla

Tässä työssä löydetty ratkaisut ovat parempia kuin kirjallisuudessa esitetyt ratkaisut. Tuloksista huomataan muutaman sauvan pinta-alan olevan alarajalla, joka viittaa siihen, että ristikon alkuperäinen topologia ei ole optimaalinen. Kuvasta 5.5 huomataan, että SLP:n avulla ratkaistussa tehtävässä solmut 4 ja 8 ovat hyvin lähellä toisiaan, mutta MISLP:n ratkaisussa nämä solmut ovat selkeästi erillään. Tämä tuo hyvin esille diskreetin tehtävän haastavuuden, sillä pienikin muutos solmujen sijainneissa saattaa pakottaa sauvan dis-



Kuva 5.5. 15-sauvan ristikon optimiratkaisut

kreetin katalogin mukaiseen suurempaan poikkileikkaukseen. Jatkuvan tehtävän (SLP) ja diskreetin tehtävän (MISLP) optimitulokset ovat hyvin lähellä toisiaan. Suurimmat erot jatkuvan ja diskreetin ratkaisun välillä ovat ristikon päädyn sauvojen pinta-aloissa sekä solmukoordinaateissa.

Suuren osan evaluoinneista muodostaa derivaatan numeerinen määrittäminen, joka suoritetaan jokaisen muuttujan suhteen. Suunnittelumuuttujien lukumäärän kasvaessa, myös optimointiin käytettävien analyysien määrä kasvaa. FEM-analyysien lukumäärä onkin lähes nelinkertainen 10-sauvan ristikkoon verrattuna. 15-sauvan ristikossa on lähes kaksi kertaa niin monta suunnittelumuuttujaa kuin 10-sauvan ristikossa ja tehtävä on haastavampi, mikä lisää analyysien lukumäärää.

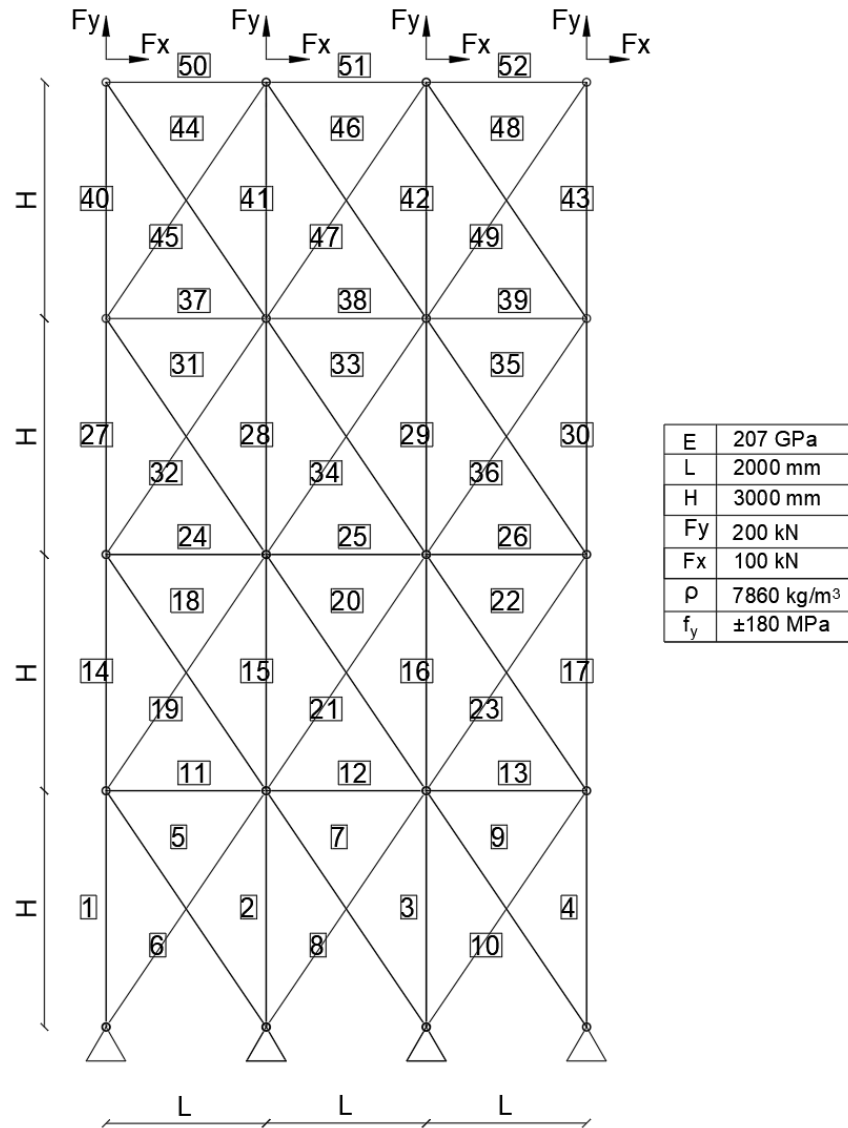
5.3 52-sauvan ristikko

52-sauvan ristikossa usealle sauvalle käytetään samaa suunnittelumuuttujaa. Tehtävänä on minimoida kuvan 5.6 mukaisen tasorakenteen massa. Jokaisen kerroksen pysty-, vaaka- ja vinosauvat muodostavat oman ryhmänsä, joille käytetään samaa suunnittelumuuttujaa. Suunnittelumuuttujina käytetään sauvojen pinta-aloja ja rajoitusehtoina sauvojen jännityksiä. Sauvajako on esitetty tarkemmin optimitulosten yhteydessä taulukossa 5.4.

Tehtävä matemaattisessa muodossa

$$\begin{aligned}
 \min \rho \sum_{i=1}^{52} L_i x_i \\
 \frac{|N_i(\mathbf{x})|}{f_y A_i} - 1 \leq 0 \\
 x_i \in \mathbf{A}_{52} \\
 \mathbf{x} = [A_1, A_2, \dots, A_{12}]
 \end{aligned} \tag{5.6}$$

Vezvari et. al [34] käyttivät tehtävän ratkaisemiseen metaheuristista algoritmia, jonka inspiraationa toimivat urheiluturnaukset (Numbers Cup Optimization, NCO). Algoritmi muis-



Kuva 5.6. 52-sauvan ristikko ja tehtävässä käytettyjen parametrien arvot

tuttaa hieman parveilualgoritmia (Particle Swarm Optimization, PSO), missä partikkelit liikkuvat joukon parhaan ja oman parhaan tuloksensa suuntaan satunnaisen askeleen verran.

Do et. al [5] kehittivät populaatiopohjaisen metaheuristisen muunnetun symbiottisen organismien hakualgoritmin (modified Symbiotic Organisms Search, mSOS), jonka avulla ristikko optimoitiin. Algoritmi on luonteeltaan yhdistelmä parveilu- ja geneettistä algoritmia. Heidän tuloksensa näyttävät hyviltä, mutta heidän tuloksissa esittämä ristikon kokonaispaino 1899.654 kg, ei vastaa optimituloksen arvoilla laskettua ristikon painoa. Taulukossa 5.4 on esitetty heidän optimituloksen mukaan laskettu ristikon massa.

Kirjallisuudesta löytyvä tehtävän paras tulos löytyy Li et. al julkaisusta [28]. Heidän ratkaisuaan ei tosin voi pitää luotettavana, sillä optimituloksen mukaiset pinta-alat eivät kuulu tehtävässä yleisesti käytettyyn profiilikatalogiin eivätkä he esitä julkaisussaan optimointiin käytettyä katalogia. Tuloksissa esitetty ristikon kokonaispaino on myös 4 kg pienempi

kuin heidän optimituloksen avulla laskettu ristikon paino. Kaiken lisäksi ratkaisu on myös epäkäypä, sillä sauvan 26 vetojännityksen rajoitusehto ylittyy 0.001, joka ei ole sallituissa toleransseissa.

Tässä työssä ristikko optimoidaan 15-sauvan ristikon tapaan jatkuvilla- ja diskreeteillä muuttujilla käyttäen SLP- ja MISLP -algoritmeja. Aloituspisteinä käytetään kokeilemalla löydettyä pienintä käypää ratkaisua

$$A_i = 3303.219, \quad i \in [1, 12] \quad (5.7)$$

Tehtävän tulokset on esitetty kootusti taulukossa 5.4. Jatkuvan tehtävän ratkaisu on huomattavasti diskreettiä ratkaisua parempi. Määrääväksi rajoitusehdoksi tulee sauvan 10 puristus. MISLP -algoritilla ratkaistun tehtävän määräävän rajoitusehdon arvo on $75e^{-8}$, joka on suurempi kuin 0, mutta se on sallituissa toleransseissa, joten ratkaisu hyväksytään.

Diskreetissä tehtävässä suuri osa ajasta kuluu linearisoidun sekalukutehtävän ratkaisemiseen. Käytettävien pinta-alojen arvojen vaihteluväli on niin suuri, että pienilläkin muutosrajojen arvoilla alitehtävään luodaan monta uutta binäärimuuttujaa. Muutosrajojen esittäminen suhteena pienimmän ja suurimman arvon vaihteluväliin ei siis välttämättä ole optimaalisin ratkaisu rajoittaa MISLP:n alitehtävässä käytettäviä diskreettejä arvoja.

Taulukko 5.4. 52-sauvan ristikon optimitulokset

Muuttuja	Sauvat	NCO [34]	mSOS [5]	SLP	MISLP
A_1	1-4	4658.055	4658.06	4416.50	4658.055
A_2	5-10	1161.288	1161.29	1114.24	1161.288
A_3	11-13	494.193	494.193	239.34	388.386
A_4	14-17	3303.219	3303.22	3392.92	3303.219
A_5	18-23	939.998	940	863.14	939.998
A_6	24-26	494.193	506.451	218.99	506.451
A_7	27-30	2238.705	2283.71	2310.13	2238.705
A_8	31-36	1008.385	1008.39	956.33	1008.385
A_9	37-39	494.193	388.386	247.21	506.451
A_{10}	40-43	1283.868	1283.87	1317.85	1283.868
A_{11}	44-49	1161.288	1161.29	1052.29	1161.288
A_{12}	50-52	494.193	506.451	417.09	506.451
Analyyseiden lkm	-	6510	7950	112	2878
Paino (kg)	1-52	1902.61	1903.02	1809.49	1899.35

Kaveh et. al. [21] löysivät tehtävälle huomattavasti muita paremman tuloksen, 1611.77 kg. Muista ratkaisuista poiketen heidän ratkaisussaan käytettiin eri sauvaryhmittelyä, jo-

ten tulos ei ole vertailukelpoinen muiden tulosten kanssa. Tämä tulos on kuitenkin hyvä muistutus siitä, että tehtävän asettelulla on todella suuri vaikutus lopputulokseen.

5.4 Tulosten yhteenveto

Tässä työssä optimoitiin kolme kirjallisuudessa yleisesti käytettyä rakennetta ja verrattiin tuloksia aiemmin löydettyihin optimituloksiin. Tulostaulukoissa on esitetty iteraatioiden lukumäärän sijaan vaadittavien FEM-analysien lukumäärä. Jokainen algoritmi tekee erilaisia asioita yhden iteraation aikana, joten iteraatioiden välinen vertailu ei ole välttämättä järkevää. FEM-analysien lukumäärää sekä laskenta-aikaa ei pystytty vertailemaan metaheurististen ja determinististen menetelmien välillä, sillä julkaisuissa ei oltu näitä esitetty.

10-sauvan ristikon optimointitehtävä ratkaistiin kaksivaiheisella tekniikalla. Ensimmäisessä vaiheessa tehtävä relaxsoitiin ja ratkaistiin SLP-algoritilla jatkuvien muuttujien avulla. Tämän jatkuvan tehtävän ratkaisun ympäriltä valittiin lähimmät diskreetit arvot euklidisen etäisyyden avulla ja tätä pistettä käytettiin aloituspisteenä toisen vaiheen diskreetissä tehtävässä, joka ratkaistiin VNS:n avulla. SLP päättyi samaan tulokseen lähes aina aloituspisteestä ja muuttujarajoista huolimatta. Diskreetissä tehtävässä päädyttiin kirjallisuudessa aiemmin löydettyyn optimitulokseen 2489,95 kg.

Determinististen algoritmien toimintaa tutkittiin myös geometriaoptimoinnissa 15-sauvan ristikon optimointitehtävässä. Tässä tehtävässä tavoite on optimoida 15-sauvaisen ristikon geometria sekä sauvojen profiilit. Tehtävä ratkaistiin SLP:llä jatkuvien muuttujien avulla ja ratkaisua verrattiin kirjallisuudesta löydettyjen tulosten lisäksi MISLP -algoritilla ratkaistuun diskreettien pinta-alamuuttujien sekä jatkuvien geometriamuuttujien avulla esitetyn tehtävän ratkaisuun. MISLP -algoritilla ongelmaksi muodostui tilanne jatkuvan tehtävän ratkaisun lähellä, sillä pienikin muutos solmujen sijainneissa saattoi pakottaa sauvan diskreetin katalogin mukaiseen suurempaan poikkileikkaukseen. Tässä työssä löydetty diskreetti tulos 32,06 kg on parempi kuin Shohaee et. al [32] aiemmin löytämä tulos 33,56 kg.

52-sauvan ristikon tehtävässä sauvat on ryhmitelty omiin ryhmiinsä ja jokaiselle ryhmän sauvalle käytetään samaa suunnittelumuuttujaa. Tehtävä ratkaistiin sekä jatkuvien, että diskreettien muuttujien avulla. Jatkuvassa tehtävässä SLP päättyi optimitulokseen 1809,49 kg vain 112 FEM-analysin jälkeen. MISLP -algoritmi vaati 2878 FEM-analyysiä diskreetin optimituloksen löytämiseen. Tässä työssä löydetty diskreetti optimitulos 1899,35 kg on parempi kuin Vezvari et. al [34] aiemmin löytämä optimitulos 1902,61 kg. MISLP -algoritilla löydetty optimitulos rikkoo hieman rajoitusehtoja, mutta tulos on kuitenkin sallituissa toleranssissa, joten ratkaisua voidaan pitää käypänä.

Näiden tehtävien perusteella voidaan todeta determinististen algoritmien soveltuvan rakenteiden diskreettiin optimointiin. Deterministisiä algoritmeja voidaan hyödyntää kaksivaiheisessa menetelmässä relaxsoidun, jatkuvien muuttujien tehtävän ratkaisemiseen tai

tehtävä voidaan ratkaista suoraan diskreeteillä muuttujilla binääriformulointia hyödyntäen. Relaksoitu, jatkuvan muuttujan tehtävä ratkeaa nopeasti ja ratkaisun ympäriltä pystyy muodostamaan hyvän diskreetin alkuarvauksen ratkaisun diskretointia varten.

6 RISTIKON OPTIMOINTI

Tässä luvussa tarkastellaan symmetrisen, kattokannattimena käytettävän tasoristikon optimointia. Ristikko valmistetaan neliöputkista ja sauvat mitoitetaan standardin SFS-EN 1993-1-1 [6] ja liitokset standardin SFS-EN 1993-1-8 [7] mukaisesti. Kohdefunktiona käytetään ristikon massaa ja rajoitusehdot muodostetaan eurokoodien mukaisten kestävyys- sekä geometriaehtojen avulla.

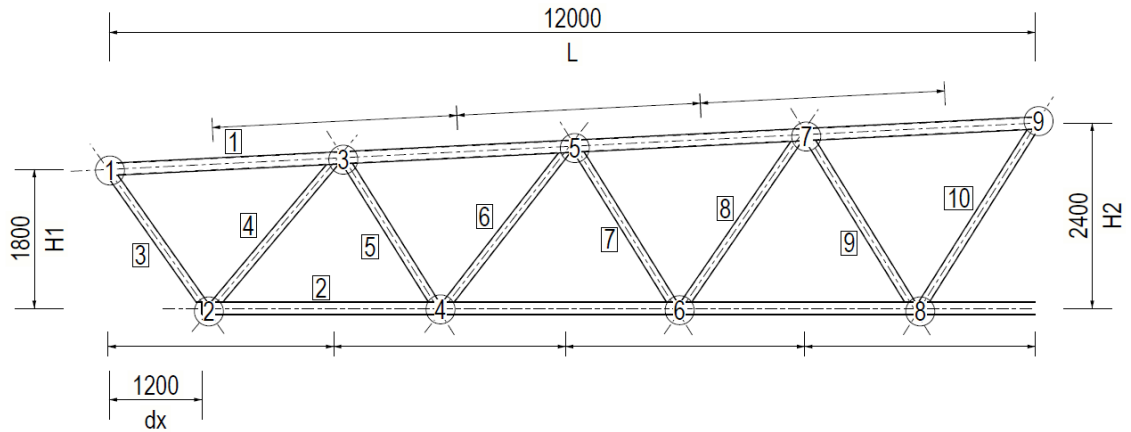
6.1 Tehtävän formulointi

Tässä työssä tarkastellaan symmetrisen 24 metriä pitkän tasoristikon optimointia. Uumasauvojen lukumäärä, 16 kappaletta, pidetään vakiona optimoinnin aikana. Ristikko optimoidaan kahdessa vaiheessa: ensimmäisessä vaiheessa ristikon sauvojen poikkileikkausten mitat sekä geometria optimoidaan jatkuvien muuttujien avulla ja toisessa vaiheessa geometria lukitaan ja sauvojen profiilit diskretoidaan käyttäen indeksimuuttujia. Kohdefunktiona käytetään ristikon massaa. Ristikon mitoille käytettävät parametrit sekä liitosten ja sauvojen numerointi on esitetty kuvassa 6.1.

Tehtävä kootusti

$$\begin{aligned}
 \min f(\mathbf{x}) &= \rho \sum_{i=1}^n L_i(\mathbf{x}_G) A_i(\mathbf{x}_M) \\
 g^\delta(\mathbf{x}_M, \mathbf{x}_G) &\leq 0 \\
 g_i^{CS}(\mathbf{x}_M, \mathbf{x}_G) &\leq 0 \\
 g_i^{MS}(\mathbf{x}_M, \mathbf{x}_G) &\leq 0 \\
 g_j^{JS}(\mathbf{x}_M, \mathbf{x}_G) &\leq 0 \\
 g_j^{JG}(\mathbf{x}_M, \mathbf{x}_G) &\leq 0
 \end{aligned} \tag{6.1}$$

missä sauvan pituus L riippuu geometriamuuttujista \mathbf{x}_G , pinta-ala A riippuu mitoitusmuuttujista \mathbf{x}_M ja teräksen tiheydelle käytetään arvoa $\rho = 7850 \text{ kg/m}^3$. Rajoitusehtoina on ristikon taipuma g^δ , sauvan poikkileikkauksen kestävyys g_i^{CS} sekä stabiilisuus g_i^{MS} ja liitoksen kestävyys g_j^{JS} sekä g_j^{JG} geometriaehdot. Rajoitusehdot on esitetty tarkemmin sauvojen mitoitus- ja stabiilisuutta sekä liitoksen kestävyyslaskentaa käsittelevissä luvuissa.



Kuva 6.1. Ristikon alkumitat sekä liitosten ja sauvojen numerointi

6.1.1 Suunnittelumuuttujat

Sauvojen suorakaideputkiprofiilien jatkuvina muuttujina käytetään poikkileikkauksen korkeutta h sekä seinämäpaksuutta t . Muuttujarajoina käytetään paarteille ja uumasauvoille seuraavia arvoja

$$\begin{aligned}
 100 &\leq h_C \leq 200 \\
 5 &\leq t_C \leq 10 \\
 50 &\leq h_W \leq 150 \\
 3 &\leq t_W \leq 6
 \end{aligned}
 \tag{6.2}$$

missä alaindeksi C viittaa paarteeseen ja W uumasauvaan.

Indeksimuuttujille käytettävät arvot valitaan suoraan SSAB:n mukaisesta katalogista. Katalogin arvot löytyvät liitteestä A. Profiilivalikoimasta poistetaan kaikki poikkileikkausluokkiin 3 ja 4 kuuluvat profiilit sekä hitsattavuuden takia seinämäpaksuudeltaan pienemmät kuin 3 mm profiilit. Paarteille käytettävästä listasta rajataan vielä pois kaikki profiilit joiden leveys on pienempi kuin 100 mm ja seinämäpaksuus on vähemmän kuin 5 mm. Käytettävissä olevat profiilit järjestetään pinta-alan mukaan kasvavaan järjestykseen.

Poikkileikkauksen ominaisuuksiin vaikuttavat muuttujavektorit ovat muotoa

$$\begin{aligned}
 \mathbf{x}_{\text{MI}} &= [i_1, i_2, \dots, i_n] \\
 \mathbf{x}_{\text{MJ}} &= [h_1, t_1, h_2, t_2, \dots, h_n, t_n]
 \end{aligned}
 \tag{6.3}$$

missä alaindeksit I ja J viittaavat indeksi- ja jatkuviin muuttujiin.

Geometriamuuttujina käytetään liitosten lokaaleja sijainteja, harjakorkeutta sekä alapaarteen ja yläpaarteen välistä etäisyyttä tuella. Liitokset liikkuvat paarteella siten, että liitosten solmupisteitä voidaan siirtää molempiin suuntiin puolet alkuperäisestä solmujaosta. Näin varmistetaan, että diagonaalit eivät risteä optimoinnin aikana.

Toinen vaihtoehto olisi kirjoittaa uusi rajoitusehto liitosten risteämiseksi. Tämä saattaa kuitenkin johtaa tilanteeseen missä liitokset risteävät rajoitusehdosta huolimatta ja aiheuttaa numeerisia ongelmia, kuten paarteen elementin pituuden lyhenemisen nolaksi, jolloin FEM-analyysia ei pystytä suorittamaan nolalla jaon seurauksena.

Alapaarteen etäisyys tuesta määräytyy alapaarteen ulomman liitoksen sijainnin mukaan. Ristikon korkeuden muuttujarajoille käytetään seuraavia arvoja

$$\begin{aligned} 900 &\leq H_1 \leq 2700 \\ 1800 &\leq H_2 \leq 3600 \end{aligned} \quad (6.4)$$

6.1.2 Ristikon rakennemalli

Ristikko tuetaan nivelellisesti yläpaarteiden päistä. Jokainen uumasauva mallinnetaan yhtenä palkkielementtinä ja poikkileikkauksen kestävyys tarkastetaan uumasauvan molemmissa päissä. Uumasauvojen ja paarteiden väliset liitokset oletetaan nivelellisiksi ja ne mallinnetaan jäykkien epäkeskisyyselementtien avulla, jotta liitoksessa vaikuttavat voimat saadaan tarkasti määritettyä. Jäykät epäkeskisyyselementit on esitetty kuvassa 6.4. Yläpaarteiden välinen liitos harjalla oletetaan jäykäksi.

Paarressauvoille mallinnetaan kaksi palkkielementtiä jokaiselle liitosvälille, jotta momenttien ja aksiaalisten voimien yhteenvaikutuksen tarkastaminen on tarkempaa. Paarressauvojen poikkileikkauksen kestävyys tarkistetaan $3m$ pisteessä, missä m on liitosten lukumäärä.

Yläpaarteeseen vaikuttaa tasainen kuorma $q = 23.5kN/m$, joka vastaa suuruudeltaan kattorakenteen, ripustusten ja lumen aiheuttamaa kuormaa. Ristikkorakenteen omaa painoa ei oteta huomioon. Materiaalina kaikille sauvoille käytetään S355J2H, jonka materiaaliominaisuudet on esitetty taulukossa 6.1.

Taulukko 6.1. S355J2H teräksen materiaaliominaisuudet

E	210 000 MPa
G	80769 MPa
f_y	355 MPa
f_u	510 MPa
ρ	7850 kg / m ³

6.2 Rajoitusehdot

Tehtävän rajoitusehtoina käytetään eurokoodin mukaan laskettuja sauvan ja liitosten kestävyksiä. Eurokoodissa kestävyudet esitetään lineaarisessa muodossa

$$E_d \leq R_d. \quad (6.5)$$

missä E_d on mitoituskuorma ja R_d eurokoodin mukaan laskettu kestävyys.

Tässä työssä rajoitusehdot esitetään suhteina, joka yhtenäistää ulkoisen sakkofunktion käyttäytymistä. Kaikki rajoitusehdot on muotoa

$$\frac{E_d}{R_d} - 1 \leq 0. \quad (6.6)$$

Sauvojen ja liitosten kestävyysien lisäksi rajoitusehtona on ristikon kokonaistaipuma

$$g^\delta = \frac{0.75|\delta_i(\mathbf{x})|}{\delta_{max}} - 1 \leq 0 \quad (6.7)$$

missä ristikon taipuman arvona käytetään 75 % murtorajatilan kuormien avulla lasketusta taipumasta ja maksimitaipuman arvona $\delta_{max} = L/150$, missä L on kuvan 6.1 mukainen etäisyys tuelta ristikon harjalle.

6.2.1 Poikkipintasuureiden määrittäminen

Sauvan poikkipintasuureet riippuvat suunnittelumuuttujista. Käytettäessä indeksimuuttujia, voidaan poikkipintasuureet lukea suoraan taulukosta valitun indeksin kohdalta, kun taas jatkuvilla muuttujilla poikkipintasuureet tulee laskea joka iteraatiokierroksella.

Neliöputkiprofiilin pinta-ala lasketaan kaavalla

$$A = 4t(h - 2r_0) + \pi(r_0^2 - (r_0 - t)^2) \quad (6.8)$$

missä h ja t ovat poikkileikkauksen korkeus sekä seinämän paksuus ja pyöristyssäde r_0 määräytyy paloittain määritellyn lineaarisen funktion avulla

$$r_0 = \begin{cases} 2t & t \leq 6 \\ 2.5t & 6 < t \leq 10 \\ 3t & t > 10 \end{cases} \quad (6.9)$$

Pyöristyssäteen epäjatkuvuus saattaa aiheuttaa haastetta optimointialgoritmille raja-arvojen läheisyydessä, sillä poikkileikkauksen pinta-ala pienenee tai suurenee siirryttäessä paloittain määritellyn pyöristyssäteiden funktiosta toiseen. Tästä voi seurata tuloksen muut-

tuminen epäkäyväksi, jolloin algoritmit jotka eivät pysty käsittelemään epäkäyviä pisteitä lopettavat iteroinnin kesken ja päätyvät näin lokaaliin minimiin.

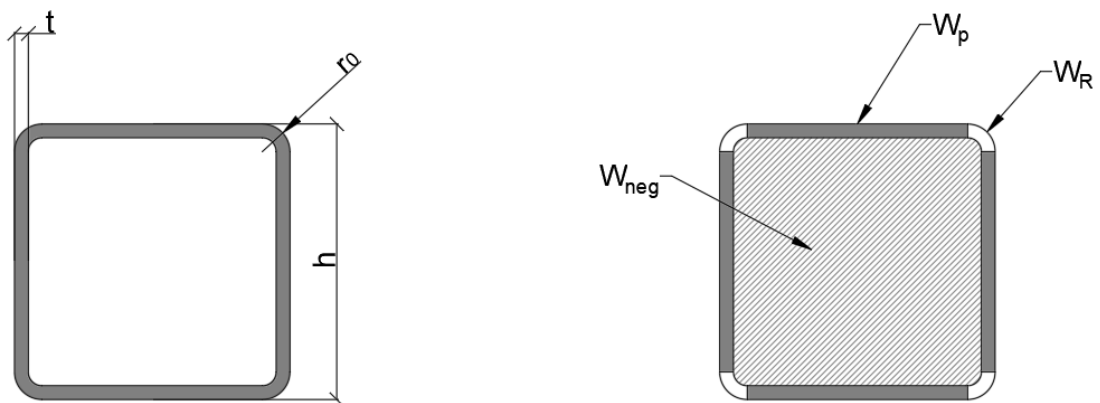
Kun seinämän paksuus on vakio, lasketaan leikkauspinta-ala seuraavasti

$$A_v = \frac{A}{h}. \quad (6.10)$$

Plastinen taivutusvastus saadaan laskettua seuraavasti

$$\begin{aligned} W_p &= t \left(2(0.5h - r_0)^2 + (h - 2r_0)(h - t) \right) \\ W_r &= \frac{r_0^2}{3} (4r_0 + \pi(1.5h - 3r_0)) \\ W_{neg} &= \frac{1}{3} (r_0 - t)^2 (4r_0 - 4t + \pi(1.5h - 3r_0)) \\ W_{pl} &= W_p + W_r - W_{neg} \end{aligned} \quad (6.11)$$

missä W_p on suorien taso-osien, W_r pyöristysten ja W_{neg} tyhjän tilan plastinen taivutusvastus kuvan 6.2 mukaisesti.



Kuva 6.2. Neliöputkiprofiilin dimensiot ja plastisen taivutusvastuksen määrittäminen

Neliöputkiprofiilin neliömomentti lasketaan samaan tapaan, kuin plastinen taivutusvastus

$$\begin{aligned} I_0 &= 2t^3 \left(\frac{h}{12} - \frac{r_0}{6} \right) + 2t \left(\frac{h}{2} - \frac{t}{2} \right)^2 (h - 2r_0) + \frac{t}{6} (h - 2r_0)^3 \\ I_r &= 4r_0^4 \left(-\frac{4}{9\pi} + \frac{\pi}{16} \right) + \pi r_0^2 \left(\frac{h}{2} - r_0 + \frac{4r_0}{3\pi} \right)^2 \\ I_{neg} &= 4 \left(-\frac{4}{9\pi} + \frac{\pi}{16} \right) (r_0 - t)^4 + \pi (r_0 - t)^2 \left(\frac{h}{2} - r_0 + \frac{1}{\pi} \left(\frac{4r_0}{3} - \frac{4t}{3} \right) \right)^2 \\ I &= I_0 + I_r - I_{neg} \end{aligned} \quad (6.12)$$

missä I_0 on suorien taso-osien, I_r pyöristysten ja I_{neg} tyhjän tilan neliömomentti.

6.2.2 Poikkileikkauksen kestävyys

Koska putket hitsataan toisiinsa kiinni, lasketaan poikkileikkauksen vetokestävyys bruttopoikkileikkaukselle plastisuusteorian mukaan

$$N_{t,Rd} = \frac{Af_y}{\gamma_{M0}} \quad (6.13)$$

missä A on poikkileikkauksen pinta-ala, f_y on käytettävän teräslaadun myötölujuus ja osavarmuuskerroin $\gamma_{M0} = 1$.

Poikkileikkausluokissa 1,2 ja 3 puristuskestävyyden mitoitusarvo tasaiselle puristukselle lasketaan seuraavasti

$$N_{c,Rd} = \frac{Af_y}{\gamma_{M0}} \quad (6.14)$$

Teräspoikkileikkauksen plastisuusteorian mukainen leikkauskestävyys lasketaan kaavasta

$$V_{c,Rd} = \frac{A_v f_y / \sqrt{3}}{\gamma_{M0}} \quad (6.15)$$

Taivutuskestävyyden mitoitusarvo yhden pääjäyhysakselin suhteen lasketaan poikkileikkausluokissa 1 ja 2 plastisuusteorian mukaisesti

$$M_{c,Rd} = M_{pl,Rd} = \frac{W_{pl} f_y}{\gamma_{M0}} \quad (6.16)$$

missä W_{pl} on plastinen taivutusvastus.

Taivutusmomentin ja normaalivoiman yhteisvaikutus on tarkistettava silloin, kun poikkileikkauksen käyttöaste normaalivoiman suhteen on yli 25 %.

$$\begin{aligned} n &\geq 0.25 \\ n &= \frac{N_{Ed}}{N_{pl,Rd}} \end{aligned} \quad (6.17)$$

Yhteisvaikutuksen seurauksena, taivutusvastus on nyt muotoa

$$M_{N,Rd} = \min(M_{pl,Rd}, M_{pl,Rd}(1 - n)/(1 - 0.5a_w)) \quad (6.18)$$

missä kerroin a_w lasketaan kaavalla

$$a_w = \min\left(0.5, \frac{A - 2ht}{A}\right) \quad (6.19)$$

Kaikilla neliöputkiprofiileilla kerroin $a_w = 0.5$, joten yhteisvaikutus voidaan tarkistaa vaikka aksiaalisen voiman suuruus ei täyttäkään edellä esitettyä ehtoa.

Taivutuksen ja leikkausvoiman yhteisvaikutus on tarkistettava, jos leikkausvoiman arvo on yli puolet plastisen leikkauskestävyyden arvosta ja leikkauslommahdus ei pienennä kestävyyttä.

$$V_{Ed} \geq 0.5V_{pl,Rd} \quad (6.20)$$

Yhteisvaikutuksen seurauksena taivutuskestävyyttä pienennetään seuraavasti

$$M_{V,Rd} = \left(W_{pl} - \frac{\rho A_v^2}{8t} \right) \frac{f_y}{\gamma_{M0}} \leq M_{pl,Rd} \quad (6.21)$$

missä pienennyskerroin ρ lasketaan kaavalla

$$\rho = \left(\frac{2V_{Ed}}{V_{pl,Rd}} - 1 \right)^2 \quad (6.22)$$

Tässä työssä yhteisvaikutus tarkistetaan joka tilanteessa käyttämällä pienennyskertoimelle arvoa 0, silloin kun kaavan 6.20 mukainen ehto ei toteudu.

Taivutuksen, normaalivoiman ja leikkausvoiman yhteisvaikutus pienentää poikkileikkauksen taivutuskestävyyttä seuraavasti

$$M_{N,V,Rd} = M_{V,Rd} \quad \text{kun} \quad N_{Ed} \leq \frac{0.5A_{w,red}f_y}{\gamma_{M0}} \quad \text{ja} \quad n_V \leq 0.25 \quad (6.23)$$

muulloin

$$M_{N,V,Rd} = k \cdot M_{V,Rd} \quad (6.24)$$

missä

$$\begin{aligned} k &= \min \left(\frac{1 - n_V}{1 - 0.5a_V}, 1 \right) \\ n_V &= \frac{N_{Ed}}{N_{V,Rd}} \\ a_V &= \min \left(\frac{A_{w,red}}{A_{tot,red}}, 0.5 \right) \\ A_{w,red} &= (1 - \rho)(A - 2bt) \\ A_{tot,red} &= A - \rho(A - 2bt) \\ N_{V,Rd} &= \frac{A_{tot,red}f_y}{\gamma_{M0}} \end{aligned} \quad (6.25)$$

Teräspoikkileikkaukset luokitellaan neljään eri poikkileikkausluokkaan puristettujen osien leveys-paksuussuhteiden avulla. [6] Tässä työssä sauvojen poikkileikkausluokka rajoitetaan luokkaan 2, jotta voidaan käyttää plastisuusteorian mukaisia kestävyyyksiä ja liitok-

sen kiertymisjäykkyydestä aiheutuvia sekundäärisiä momentteja ei tarvitse huomioida. Kaikki sauvat luokitellaan täysin puristettuina, joka on varmallalla puolella oleva oletus. Puristetun poikkileikkauksen tulee nyt täyttää ehto

$$\frac{c}{t} \leq 38\epsilon \quad (6.26)$$

missä $c = h - 2r_0$ ja $\epsilon = \sqrt{235/f_y}$.

Käytettäessä indeksimuuttujia voidaan käytössä olevat profiilit rajata valmiiksi kuuluvan luokkaan 2, joten rajoitusehtoa ei tarvitse tarkistaa. Jatkuvien muuttujien ongelmassa pyöristyssäde r_0 aiheuttaa rajoitusehdolle epäjatkuvuutta, joten on valittava minkä funktion suhteen rajoitusehto kirjoitetaan. Tässä työssä käytetään $r_0 = 2t$, joten poikkileikkauksiluokitus voidaan esittää muodossa

$$h - 3t \leq 38\epsilon. \quad (6.27)$$

Poikkileikkauksen kestävyys ja poikkileikkaukseluokan rajoitusehdot voidaan nyt esittää kootusti

$$g_i^{CS}(\mathbf{x}) := \begin{cases} g_{ij}^T & = \frac{N_{Ed,i}(\mathbf{x})}{N_{t,Rd,i}(\mathbf{x})} - 1 \leq 0 \\ g_{ij}^C(\mathbf{x}) & = \frac{-N_{Ed,i}(\mathbf{x})}{N_{c,Rd,i}(\mathbf{x})} - 1 \leq 0 \\ g_{ij}^V(\mathbf{x}) & = \frac{|V_{Ed,i}(\mathbf{x})|}{V_{c,Rd,i}(\mathbf{x})} - 1 \leq 0 \\ g_{ij}^M(\mathbf{x}) & = \frac{|M_{Ed,i}(\mathbf{x})|}{M_{c,Rd,i}(\mathbf{x})} - 1 \leq 0 \\ g_{ij}^{MN}(\mathbf{x}) & = \frac{|M_{Ed,i}(\mathbf{x})|}{M_{N,Rd,i}(\mathbf{x})} - 1 \leq 0 \\ g_{ij}^{MV}(\mathbf{x}) & = \frac{|M_{Ed,i}(\mathbf{x})|}{M_{V,Rd,i}(\mathbf{x})} - 1 \leq 0 \\ g_{ij}^{MNV}(\mathbf{x}) & = \frac{|M_{Ed,i}(\mathbf{x})|}{M_{N,V,Rd,i}(\mathbf{x})} - 1 \leq 0 \\ g_i^{PLL}(\mathbf{x}) & = \frac{h_i - 3t_i}{38\epsilon} - 1 \leq 0 \end{cases} \quad (6.28)$$

missä alaindeksi i viittaa tarkasteltavaan sauvaan ja j sauvan elementin solmupisteeseen.

6.2.3 Sauvan stabiilisuus

Puristetun sauvan nurjahduskestävyyden mitoitusarvo poikkileikkausluokille 1,2 ja 3 lasketaan kaavasta

$$N_{b,Rd} = \frac{\chi A f_y}{\gamma_{M1}} \quad (6.29)$$

missä χ on kyseeseen tuleva nurjahduskestävyyden pienennystekijä ja osavarmuuskerroin $\gamma_{M1} = 1$.

Aksiaalisesti puristetun sauvan muunnettua hoikkuutta $\bar{\lambda}$ vastaava pinennystekijä χ lasketaan

$$\chi = \frac{1}{\Phi + \sqrt{\Phi^2 - \bar{\lambda}^2}}, \quad \chi \leq 1 \quad (6.30)$$

missä apusuure Φ lasketaan kaavalla

$$\Phi = 0.5(1 + \alpha(\bar{\lambda} - 0.2) + \bar{\lambda}^2) \quad (6.31)$$

missä α on eurokoodin mukainen nurjahduskäyrästä riippuva epätarkkuustekijä. Kylmämuovatuille neliöputkiprofiileille käytetään nurjahduskäyrän c mukaista arvoa, eli $\alpha = 0.49$ [6].

Muunnettu hoikkuus poikkileikkausluokille 1, 2 ja 3 lasketaan seuraavasti

$$\bar{\lambda} = \sqrt{\frac{A f_y}{N_{cr}}} \quad (6.32)$$

missä N_{cr} on kimmoteorian mukainen kriittinen nurjahdusvoima, joka lasketaan kaavalla

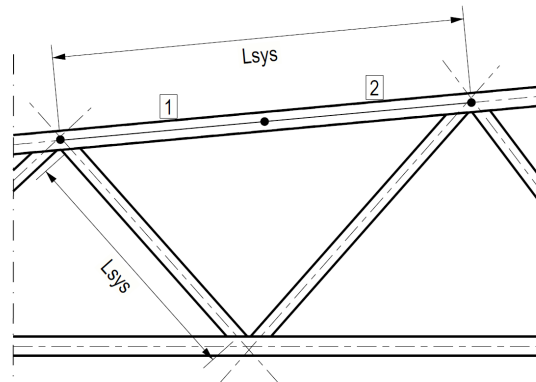
$$N_{cr} = \frac{\pi^2 EI}{L_{cr}^2} \quad (6.33)$$

missä E on materiaalin kimmokerroin, I on poikkileikkauksen neliömomentti ja L_{cr} on sauvan nurjahduspituus.

Putkiristikon paarteen nurjahduspituudeksi L_{cr} voidaan valita $0.9L_{sys}$ sekä ristikon tasossa että ristikon tasoa vastaan kohtisuorassa tasossa tapahtuvassa nurjahduksessa. Systemipituus L_{sys} ristikon tasossa on liitosten välinen etäisyys ja tasoa vastaan poikittaisten tukien välinen etäisyys. [6] Tässä työssä ristikon yläpaarre oletetaan tuettavan poikittaissuunnassa vaakasitein joka liitoksen kohdalta, joten nurjahduspituutena molempiin suuntiin voidaan käyttää $L_{cr} = 0.9L_{sys}$. Jos kattorakenteena käytetään kantavaa teräs-poimulevyä, voidaan sen olettaa tukevan ristikko tason suuntaista nurjahdusta vastaan, jolloin tason suuntaisen nurjahduksen rajoitusehto voidaan poistaa yläpaarteilta.

Uumasauvat voidaan mitoittaa ristikon tasossa tapahtuvalle nurjahdukselle käyttämällä systeempituutta pienempää nurjahduspituutta edellyttäen, että paarteet antavat tarkoituksenmukaisen tuennan ja uumasauvan kiinnitys paarteeseen on riittävän jäykkä. Tavallisissa ristikkorakenteissa uuman nurjahduspituudeksi L_{cr} ristikon tasossa voidaan valita $0.9L_{sys}$ kulmateräksiä lukuun ottamatta. [6]

Tässä työssä paarteen sekä uumasauvojen nurjahduspituutena käytetään $0.9L_{sys}$, joka on varmalla puolella oleva oletus uumasauvoille.



Kuva 6.3. Sauvojen systeempituudet ja yläpaarteen elementtijako

Yläpaarteeseen vaikuttaa yhdistetty taivutus ja aksiaalinen puristus, joten sauvan stabiilisuus näille yhdistetyille rasituksille on tarkistettava. Kestävyyteen vaikuttaa sauvan nurjahduskestävyys, kipehaduskestävyys sekä momenttipinnan muoto.

Yhteisvaikutustekijä k_{yy} lasketaan kaavalla

$$k_{yy} = C_{my} \left(1 + (\bar{\lambda}_y - 0.2) \frac{N_{Ed}}{\chi_y N_{Rk} / \gamma_{M1}} \right) \leq C_{my} \left(1 + 0.8 \frac{N_{Ed}}{\chi_y N_{Rk} / \gamma_{M1}} \right) \quad (6.34)$$

missä ekvivalentin momentin kerroin C_{my} lasketaan ristikon yläpaarteen tapauksessa kaavalla

$$C_{my} = \max(0.4, 0.1 - 0.8\alpha_s) \quad (6.35)$$

missä α_s on kenttämomentin ja itseisarvoltaan suuremman päätymomentin arvojen suhde.

Tässä työssä momenttien suhteille käytetään vakioarvoa $\alpha_s = -1$, jonka seurauksena ekvivalentin momentin kerroin saa arvon $C_{my} = 0.9$, joka on varmalla puolella oleva oletus.

Pyöreät ja neliön muotoiset rakenneputket eivät ole alttiita vääntömuodonmuutoksille, joten kiepahduksen pienennystekijälle χ_{LT} voidaan käyttää arvoa 1. Suorakaiteen mukaisille rakenneputkille, joihin kohdistuu aksiaalinen puristus ja taivutus yhden akselin suhteen, tekijälle k_{zy} voidaan käyttää arvoa 0. [6]

Sauvan stabiilisuuden rajoitusehdot kootusti

$$g_i^{MS}(\mathbf{x}) := \begin{cases} g_i^B(\mathbf{x}) & = \frac{-N_{Ed,i}(\mathbf{x})}{N_{b,Rd,i}(\mathbf{x})} - 1 \leq 0 \\ g_i^{NM}(\mathbf{x}) & = \frac{-N_{Ed,i}(\mathbf{x})}{N_{b,Rd,i}(\mathbf{x})} + k_{yy}(\mathbf{x}) \frac{|M_{Ed,i}(\mathbf{x})|}{M_{c,Rd,i}(\mathbf{x})} - 1 \leq 0 \end{cases} \quad (6.36)$$

6.2.4 Liitoksen geometria

Liitokselle asetetut geometriaehdot on esitetty standardin SFS-EN 1993-1-8 [7] taulukossa 7.8. Liitoksen epäkeskisyydestä aiheutuvia momenteja ei tarvitse ottaa liitoksen, uumasauvojen ja vedetyn paarteen mitoituksessa, jos sauvojen systeemipituudet L_{sys} sekä liitoksen epäkeskisyyden toteuttavat seuraavat ehdot [7]

$$-0.55h_0 \leq e \leq 0.25h_0 \quad (6.37)$$

$$L_{sys,i} \geq h_i \quad (6.38)$$

missä alaindeksi i viittaa kyseessä olevaan sauvaan.

Liitosten kiertymisjäykkyydestä aiheutuvat liitosten sekundääriset vaikutukset voidaan jättää huomioimatta, jos sauvan systeemipituus toteuttaa sille asetetun ehdon ja liitoksen geometria toteuttaa kyseiselle liitokselle esitetyt vaatimukset.

Tässä työssä liitokset mallinnetaan jäykkien epäkeskisyyselementtien avulla, jotta liitoksissa vaikuttavat rasitukset saadaan määritettyä tarkasti. Kuvassa 6.4 on esitetty liitoksen parametrit sekä jäykät epäkeskisyyselementit. Liitoksen keskipiste paarteella c_0 saadaan laskettua uumasauvojen keskilinjojen leikkauspisteen, epäkeskisyyden ja paartetta kohtisuoraa olevan yksikkövektorin avulla.

K-liitoksen epäkeskisyydellä tarkoitetaan uumasauvojen keskilinjojen leikkauspisteen etäisyyttä paarteen keskilinjasta. K-liitoksen epäkeskisyyden lasketaan kaavalla

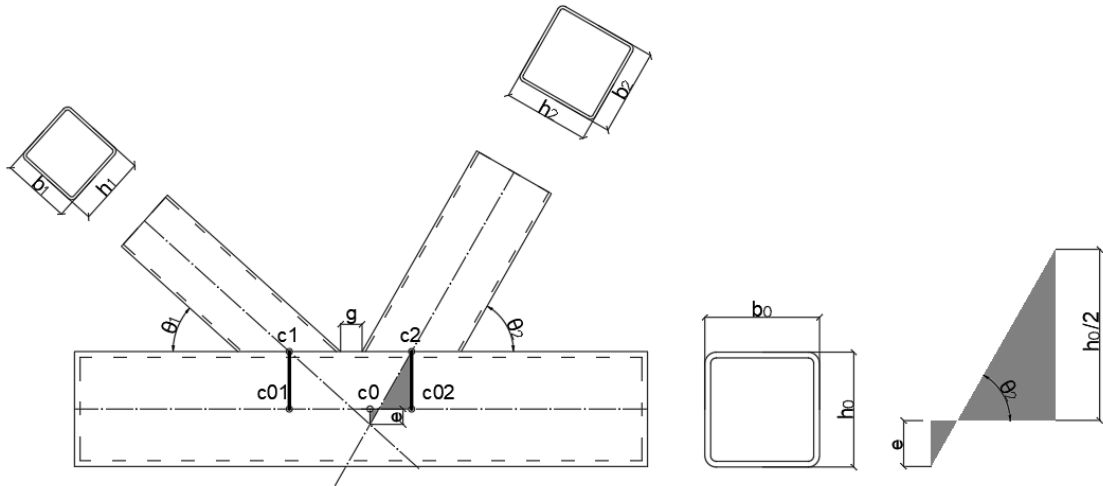
$$e = \left(\frac{h_1}{2\sin\theta_1} + \frac{h_2}{2\sin\theta_2} + g \right) \frac{\sin\theta_1 \sin\theta_2}{\sin(\theta_1 + \theta_2)} - \frac{h_0}{2}. \quad (6.39)$$

Vastaavasti vapaaväli voidaan määrittää epäkeskisyyden avulla

$$g = \left(e + \frac{h_0}{2} \right) \frac{\sin(\theta_1 + \theta_2)}{\sin\theta_1 \sin\theta_2} - \frac{h_1}{2\sin\theta_1} - \frac{h_2}{2\sin\theta_2} \quad (6.40)$$

K-liitoksen vapaavälin tulee täyttää seuraava ehto, muuten liitosta tulee käsitellä kahtena erillisenä Y-liitoksena

$$0.5(1 - \beta)b_0 \leq g \leq 1.5(1 - \beta)b_0 \quad (6.41)$$



Kuva 6.4. K-liitoksessa pisteiden $c01$ ja $c1$ sekä $c02$ ja $c2$ välille muodostetaan jäykät epäkeskisyyselementit

missä tekijä β lasketaan kaavalla

$$\beta = \frac{b_1 + h_1 + b_2 + h_2}{4b_0} \quad (6.42)$$

Liitoksen hitsattavuuden kannalta, vapaavälin tulee olla vähintään

$$g \geq t_1 + t_2 \quad (6.43)$$

Paarten ja uumasauvan välinen kulma tulee olla vähintään 30° , sillä hitsaaminen pienillä kulmilla on vaikeaa.

$$30^\circ \leq \theta_i \leq 90^\circ \quad (6.44)$$

Y-liitoksilla uumasauvan leveyden suhde paarten leveyteen on oltava vähintään 25 % ja K-liitoksilla 35 %.

$$\begin{aligned} 0.25 \leq \frac{b_i}{b_0} \leq 1.0 & \quad \text{Y-liitos} \\ 0.35 \leq \frac{b_i}{b_0} \leq 1.0 & \quad \text{K-liitos} \end{aligned} \quad (6.45)$$

Vapaavälisten N- ja K-liitosten uumasauvan ja paarten leveyden suhteen tulee täyttää myös ehto

$$\frac{b_i}{b_0} \geq 0.1 + 0.01 \frac{b_0}{t_0} \quad (6.46)$$

Liitoksen geometriaehdot kootusti Y-liitoksille

$$g_j^{JG.Y}(\mathbf{x}) := \begin{cases} g_j^{Y\theta MIN}(\mathbf{x}) & = \frac{30^\circ}{\theta_i(\mathbf{x})} - 1 \leq 0 \\ g_j^{Y\theta MAX}(\mathbf{x}) & = \frac{\theta_i(\mathbf{x})}{90^\circ} - 1 \leq 0 \\ g_j^{Ybib0MIN}(\mathbf{x}) & = \frac{0.25}{b_i(\mathbf{x})/b_0(\mathbf{x})} - 1 \leq 0 \\ g_j^{Ybib0MAX}(\mathbf{x}) & = \frac{b_i(\mathbf{x})}{b_0(\mathbf{x})} - 1 \leq 0 \end{cases} \quad (6.47)$$

sekä K-liitoksille

$$g_j^{JG.K}(\mathbf{x}) := \begin{cases} g_j^{KeMIN}(\mathbf{x}) & = \frac{-0.55h_0(\mathbf{x})}{e(\mathbf{x})} - 1 \leq 0 \\ g_j^{KeMAX}(\mathbf{x}) & = \frac{e(\mathbf{x})}{0.25h_0(\mathbf{x})} - 1 \leq 0 \\ g_j^{K\theta MIN}(\mathbf{x}) & = \frac{30^\circ}{\theta_i(\mathbf{x})} - 1 \leq 0 \\ g_j^{K\theta MAX}(\mathbf{x}) & = \frac{\theta_i(\mathbf{x})}{90^\circ} - 1 \leq 0 \\ g_j^{KgMIN_t}(\mathbf{x}) & = \frac{t_1(\mathbf{x}) + t_2(\mathbf{x})}{g_i(\mathbf{x})} - 1 \leq 0 \\ g_j^{KgMIN}(\mathbf{x}) & = \frac{0.5(1 - \beta(\mathbf{x}))b_0(\mathbf{x})}{g_i(\mathbf{x})} - 1 \leq 0 \\ g_j^{KgMAX}(\mathbf{x}) & = \frac{g_i(\mathbf{x})}{1.5(1 - \beta(\mathbf{x}))b_0(\mathbf{x})} - 1 \leq 0 \\ g_j^{Kbib0}(\mathbf{x}) & = \frac{0.1 + 0.01b_0(\mathbf{x})/b_i(\mathbf{x})}{b_i(\mathbf{x})/b_0(\mathbf{x})} - 1 \leq 0 \\ g_j^{Kbib0MIN}(\mathbf{x}) & = \frac{0.35}{b_i(\mathbf{x})/b_0(\mathbf{x})} - 1 \leq 0 \\ g_j^{Kbib0MAX}(\mathbf{x}) & = \frac{b_i(\mathbf{x})}{b_0(\mathbf{x})} - 1 \leq 0 \end{cases} \quad (6.48)$$

6.2.5 Liitoksen kestävyys

Vahvistamattomien X, Y tai T-liitoksen kestävyys lasketaan standardin SFS-EN 1993-1-8 [7] taulukon 7.11 mukaisesti.

Paarteen pinnan myötäminen

$$N_{cf,Rd} = \frac{k_n f_{y0} t_0^2}{(1 - \beta) \sin \theta_i} \left(\frac{2\eta}{\sin \theta_i} + 4\sqrt{1 - \beta} \right) / \gamma_{M5} \quad (6.49)$$

Paarteen uuman lommahdus

$$N_{cwb,Rd} = \frac{f_b t_0}{\sin \theta_i} \left(\frac{2h_i}{\sin \theta_i} + 10t_0 \right) / \gamma_{M5} \quad (6.50)$$

Uumasauvan myötäminen

$$N_{wf,Rd} = f_{yi} t_i (2h_i - 4t_i + 2b_{eff}) / \gamma_{M5} \quad (6.51)$$

Lävistysleikkaus

$$N_{ps,Rd} = \frac{f_{y0} t_0}{\sqrt{3} \sin \theta_i} \left(\frac{2h_i}{\sin \theta_i} + 2b_{e,p} \right) / \gamma_{M5} \quad (6.52)$$

Liitosten laskennassa käytettävät apusuureet saadaan laskettua seuraavien kaavojen avulla

$$\begin{aligned} \eta &= \frac{h_1}{b_0} \\ \gamma &= \frac{b_0}{2t_0} \\ f_b &= \chi f_{y0} \\ \bar{\lambda} &= 3.46 \frac{\left(\frac{h_0}{t_0} - 2 \right) \sqrt{1 / \sin \theta_i}}{\pi \sqrt{E / y_0}} \\ b_{eff} &= \frac{10}{b_0 / t_0} \frac{f_{y0} t_0}{f_{yi} t_i} b_i, \quad b_{eff} \leq b_i \\ b_{e,p} &= \frac{10}{b_0 / t_0} b_i, \quad b_{e,p} \leq b_i \\ k_n &= 1.3 - \frac{0.4n}{\beta}, \quad k_n \leq 1.0, \quad \text{vedossa } k_n = 1 \\ n &= \frac{N_{0,Ed}}{A_{0fy} / \gamma_{M5}} + \frac{M_{0,Ed}}{W_{el,0} f_{y0} / \gamma_{M5}} \end{aligned} \quad (6.53)$$

missä pienennystekijä χ lasketaan kaavan 6.22 mukaisesti ja $N_{0,Ed}$ sekä $M_{0,Ed}$ ovat paarteissa vaikuttavat itseisarvoltaan suurin normaalivoima ja taivutusmomentti.

Y-liitoksen kestävyden kaavat voidaan nyt esittää kootusti

$$g_j^{JS.Y}(\mathbf{x}) := \begin{cases} g_j^{Y,cf}(\mathbf{x}) &= \frac{|N_{Ed,j,1}(\mathbf{x})|}{N_{cf,Rd,j,1}(\mathbf{x})} - 1 \leq 0 \\ g_j^{Y,cwb}(\mathbf{x}) &= \frac{-N_{Ed,j,1}(\mathbf{x})}{N_{cwb,Rd,j,1}(\mathbf{x})} - 1 \leq 0 \\ g_j^{Y,bf}(\mathbf{x}) &= \frac{|N_{Ed,j,1}(\mathbf{x})|}{N_{bf,Rd,j,1}(\mathbf{x})} - 1 \leq 0 \\ g_j^{Y,ps}(\mathbf{x}) &= \frac{|N_{Ed,j,1}(\mathbf{x})|}{N_{bf,Rd,j,1}(\mathbf{x})} - 1 \leq 0 \end{cases} \quad (6.54)$$

K-liitoksen kestävyys lasketaan standardin SFS-EN 1993-1-8 [7] taulukon 7.12 mukai-

sesti.

Paarteen pinnan myötäminen

$$N_{cf,Rd} = \frac{8.9k_n f_{y0} t_0^2 \sqrt{\gamma}}{\sin \theta_i} \left(\frac{b_1 + b_2 + h_1 + h_2}{4b_0} \right) / \gamma_{M5} \quad (6.55)$$

Paarteen leikkaus

$$N_{v,Rd,i} = \frac{f_{y0} A_v}{\sqrt{3} \sin \theta_i} / \gamma_{M5} \quad (6.56)$$

Paarteen kestävyys normaalivoimille liitoksen vapaavälin kohdalla

$$N_{0,gap,Rd} = \left[(A_0 - A_v) f_{y0} + A_v f_{y0} \sqrt{1 - (V_{Ed}/V_{pl,Rd})^2} \right] / \gamma_{M5} \quad (6.57)$$

Uumasauvan myötäminen

$$N_{i,Rd} = f_{yi} t_i (2h_i - 4t_i + b_i + b_{eff}) / \gamma_{M5} \quad (6.58)$$

Lävistysleikkautuminen

$$N_{v,Rd} = \frac{f_{y0} t_0}{\sqrt{3} \sin \theta_i} \left(\frac{2h_i}{\sin \theta_i} + b_i + b_{e,p} \right) / \gamma_{M5} \quad (6.59)$$

Liitoksen leikkauspinta-ala lasketaan kaavalla

$$A_v = (2h_0 + \alpha b_0) t_0 \quad (6.60)$$

missä kerroin α lasketaan neliön tai suorakaiteen muotoiselle ummasauvalle seuraavasti

$$\alpha = \sqrt{\frac{1}{1 + \frac{4g^2}{3t_0^2}}} \quad (6.61)$$

Laskennassa tarvittavat muut apusuureet lasketaan samoilla kaavoilla kuin Y-liitoksen tapauksessa.

K-liitoksen kestävyys ehdot kootusti:

$$g_j^{JS.K}(\mathbf{x}) := \begin{cases} g_j^{K,cf}(\mathbf{x}) & = \frac{|N_{Ed,j,i}(\mathbf{x})|}{N_{cf,Rd,j,i}(\mathbf{x})} - 1 \leq 0 \\ g_j^{K,ps}(\mathbf{x}) & = \frac{|N_{Ed,j,i}(\mathbf{x})|}{N_{ps,Rd,j,i}(\mathbf{x})} - 1 \leq 0 \\ g_j^{K,cs}(\mathbf{x}) & = \frac{|N_{Ed,j,i}(\mathbf{x})|}{N_{cs,Rd,j,i}(\mathbf{x})} - 1 \leq 0 \\ g_j^{K,bf}(\mathbf{x}) & = \frac{|N_{Ed,j,i}(\mathbf{x})|}{N_{bf,Rd,j,i}} - 1 \leq 0 \\ g_j^{K,gap}(\mathbf{x}) & = \frac{|N_{0,Ed,j}(\mathbf{x})|}{N_{0,gap,Rd,j}(\mathbf{x})} - 1 \leq 0 \end{cases} \quad (6.62)$$

missä alaindeksi j on liitoksen indeksi ja i on liitoksen sauvan numero.

6.3 Tulokset

Aloituspisteenä käytetään paarteille profiilia SHS 160x8 ja uumasauvoille SHS 120x4, joka saadaan iteroimalla paarre- ja uumasauvojen profiileja, kunnes tehtävälle löydetään käypä ratkaisu. Ristikön liitokset on jaettu tasavälein ylä- sekä alapaarteelle. Ristikön alkugeometria on esitetty kuvassa 6.1. Paarteille ja uumasauvoille käytetään nurjahduspi- tuutena $L_{cr} = 0.9L_{sys}$.

SLP-algoritmillemme käytetään seuraavia parametrien arvoja

$$\begin{aligned} m_L &= m_U = 0.1 \\ \gamma &= 0.01 \\ \beta &= 10 \\ C &= 200000 \end{aligned} \quad (6.63)$$

VNS:n lokaalina hakualgoritmina käytetään genettistä algoritmia ja populaation kokona $n = 100$. Kahden pisteen risteytys tapahtuu 90 % todennäköisyydellä, yksilö mutatoituu 15 % todennäköisyydellä ja mutatoituessaan yksittäisen geenin arvo muuttuu nykyisessä hakuympäristössä satunnaisesti 15 % todennäköisyydellä. Populaation vanhemmiksi valitaan kaksi populaation pätevintä yksilöä. Uuteen hakuympäristöön siirrytään, jos geneettinen algoritmi ei löydä parempaa tulosta kolmen iteraatiokierroksen aikana. Haku- ympäristöä laajennetaan maksimissaan viisi kertaa jonka jälkeen optimointi lopetetaan jos parempaa tulosta ei löydetä.

Tehtävä ratkaistaan kahdessa vaiheessa. Ensimmäisessä vaiheessa ratkaistaan relax- soitu geometriaoptimointitehtävä, jossa suunnittelumuuttujina ristikön geometrialle se- kä poikkileikkausten mitoille käytetään jatkuvia muuttujia ja tehtävä ratkaistaan SLP- algoritmin avulla. Jatkuvan ratkaisun ympäriltä haetaan lähimmät diskreetit profiilit, joita

käytetään toisen vaiheen aloituspisteenä. Toisessa vaiheessa geometria pidetään muuttumattomana ja SSAB:n katalogin mukaiset profiilit pyritään löytämään VNS-algoritilla. Liitoksen vapaaväli otetaan kuitenkin mukaan optimointiin, sillä uumasauvojen muuttuminen vaikuttaa liitoksen epäkeskisyyteen, minkä seurauksena tuloksesta voi tulla epäkäypä.

6.3.1 Ristikon geometriaoptimointi

Geometriaoptimoinnissa vapaavälille asetetaan vakioarvo $g = 40mm$, jotta kaikki liitoksen kestävyys ehdot pystytään laskemaan optimoinnin aikana. Uumasauvojen poikkileikkaukset kasvavat siten, että vapaavälin arvo pysyy koko optimoinnin ajan vakiona. Mitä suurempi vapaavälin arvo sitä pienempi epäkeskisyyttä, joka tulee usein rajoittavaksi ehdoksi. Vapaaväliä voitaisiin käyttää myös suunnittelumuuttujana optimoinnin aikana, mutta se lisää tehtävän haastavuutta huomattavasti.

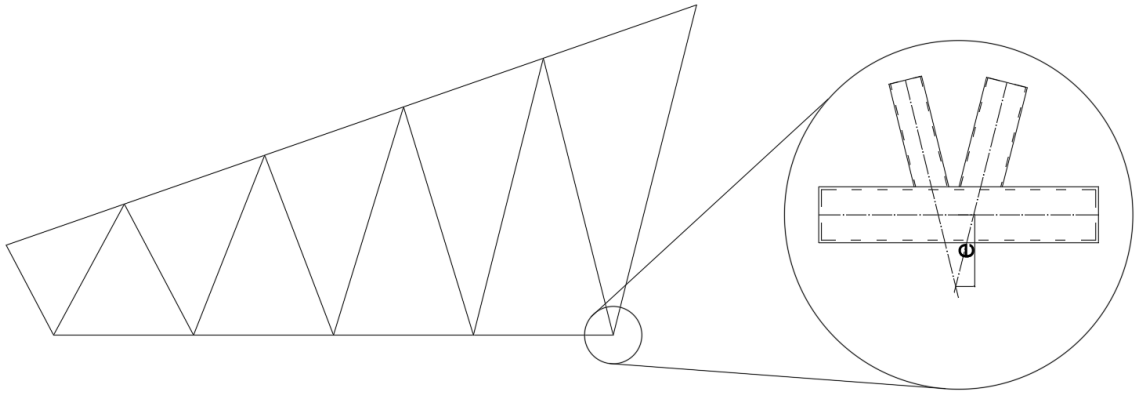
Optimoinnissa käytettävä vapaavälin vakioarvo voidaan valita käyttämällä liitoksen leveyksien suhteelle arvoa $\beta = 0.7 - 0.8$ ja laskemalla suurin minimiarvo sekä pienin maksimiarvo käyttäen paarteen korkeuden suunnittelumuuttujien rajoja. Jatkuvan tehtävän optimituloksen ei tarvitse olla käypä jotta diskreetti optimitulos voidaan pisteen ympäriltä löytää, joten vapaavälin vakioarvon tulee olla vain oikeaa suuruusluokkaa. Tässä työssä vapaavälin arvo valittiin seuraavien arvojen mukaan

$$\begin{aligned} g_{min.MAX} &= 0.5(1 - \beta)h_{0.MAX} = 20...30mm \\ g_{max.MIN} &= 1.5(1 - \beta)h_{0.MIN} = 30...45mm \end{aligned} \quad (6.64)$$

Katon kaltevuutta on myös rajoitettava, sillä suurilla kaltevuuksilla uumasauvojen kulmat ovat niin jyrkkiä, että epäkeskisyyttä ei pysy enää sallituissa rajoissa. Kuvassa 6.5 on havainnollistettu miten epäkeskisyyttä karkaa alapaarteella kauas sallituista rajoista kattokaltevuuden kasvaessa. Yläpaarteessa vaikuttava normaalivoima pienenee kun harjakorkeus kasvaa ja yläpaarteesta muodostaa huomattavan osan koko ristikkorakenteen massasta, joten optimointialgoritmin näkökulmasta harjakorkeuden kasvattaminen on järkevää, sillä paarteelle voidaan käyttää pienempää profiilia. Toinen vaihtoehto ongelman välttämiseksi on rajoittaa liittyvien uumasauvojen kulmien summaa. Tässä työssä katon kaltevuus rajoitetaan välille

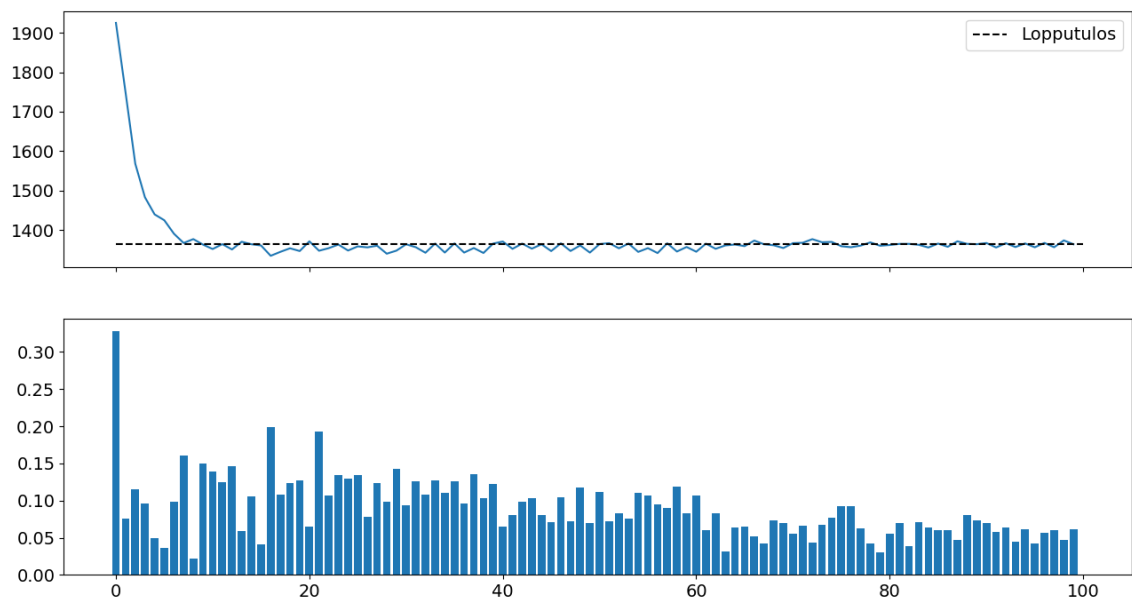
$$\begin{aligned} 1 : 40 &\leq \frac{\Delta H}{L} \leq 1 : 10 \\ \Delta H &= H_2 - H_1 \end{aligned} \quad (6.65)$$

Jatkuva tehtävä ratkaistaan käyttämällä SLP-algoritmia. Optimointi lopetetaan 100 iteraation jälkeen tai jos kahden peräkkäisen pisteen välinen etäisyys on pienempi kuin



Kuva 6.5. Suurilla kaltevuuskulmilla alapaarteen liitoksen epäkeskisyyys ei pysy sallituissa rajoissa

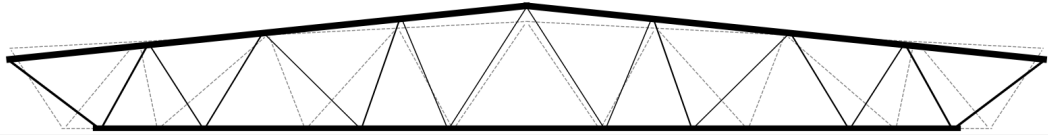
0.0001. Kuvassa 6.6 on esitetty kohdefunktion sekä rajoitusehtojen maksimiarvon kehitys optimoinnin edetessä. Kohdefunktion arvo ei muutu enää merkittävästi ensimmäisen 20 iteraatiokierroksen jälkeen. Rajoitusehtojen maksimiarvo pienenee tasaisesti lähesyttävässä 100 iteraation rajaa. Maksimirajoitusehdoksi tehtävälle muodostui yläparteen stabiilisuus taivutuksen ja puristuksen yhteisvaikutukselle.



Kuva 6.6. Jatkuvan tehtävän kohdefunktio ja rajoitusehtojen maksimiarvo iteroinnin edetessä

Kuvassa 6.7 on esitetty ristikon optimigeometria, missä viivan paksuus kuvastaa sauvan pinta-alaa. Kuvasta nähdään selkeästi miten algoritmi pyrkii lyhentämään puristettuja uumasauvoja nurjahduksen takia. Jatkuvan tehtävän poikkileikkausten optimiarvot on esitetty taulukossa 6.4.

Jatkuva ratkaisu diskretoidaan VNS-algoritmin avulla. Heurististen algoritmien satunnaisuudesta johtuen diskretointi suoritetaan 10 kertaa. Jatkuvan ratkaisun ympäriltä valitaan 10 lähintä profiilia, jotka muodostavat jokaiselle diskreetille muuttujalle alilistan käytet-



Kuva 6.7. Kattoristikon optimigeometria (yhtenäinen viiva) verrattuna alkuperäiseen geometriaan (katkoviiva)

tävissä olevista profiileista. Lähimmät profiilit valitaan euklidisen etäisyyden perusteella käyttäen parametreina poikkileikkauksen leveyttä, korkeutta sekä seinämäpaksuutta.

Jatkuvan arvon etäisyys diskreettiin profiiliin lasketaan seuraavasti

$$d_{ij} = \sqrt{(h_{ij} - \hat{h}_i)^2 + (b_{ij} - \hat{b}_i)^2 + (t_{ij} - \hat{t}_i)^2} \quad (6.66)$$

missä alaindeksi i viittaa suunnittelumuuttujaan ja j katalogin mukaisen poikkileikkauksen järjestysnumeroon.

Vaikka ristikon geometria on optimoitu jo jatkuvien muuttujien avulla, on diskreetissä tehtävässä sallittava liitoksille pieni liikkumavara sillä muuten liitoksen geometriaehtoja ei pystytä välttämättä täyttämään. Tässä työssä liitos voi siirtyä diskreetissä tehtävässä 2.5 % paarteen pituudesta eteen- tai taaksepäin. 24 metrin ristikon tapauksessa tämä tarkoittaa alapaarteen liitokselle 600 mm vaihteluväliä joka diskretoidaan 12 arvoksi, jolloin yhden askeleen pituudeksi tulee 50 mm.

Diskreetin tehtävän vapaavälin arvona aloituspisteessä käytetään $g_j = 40mm$, joka on sama kuin geometriaoptimoinnissa käytetty vakioarvo. Vapaavälin muuttujarat diskreetissä tehtävässä on

$$20 \leq g_j \leq 100 \quad (6.67)$$

joka diskretoidaan 2 mm välein.

Diskreetissä tehtävässä on siis muuttujina profiilien indeksit, diskretoitu liitoksen sijainti sekä vapaaväli. Geometriaoptimoinnin diskreetit tulokset on esitetty taulukossa 6.2. Jatkuvan ja diskreetin tehtävän ratkaisemiseksi tarvitaan yhteensä lähes 5000 FEM-analyysejä ja geometriaoptimointiin kuluu aikaa kokonaisuudessaan keskimäärin 20 minuuttia.

Ketola käytti diplomityössään [22] ristikon geometriaoptimointiin parveilualgoritmia ja sauvojen sekä liitosten mitoitus suoritettiin erillisellä mitoitusalgoritmeilla. Hänen työssään uumasauvojen nurjahduspituutenä käytettiin $L_{cr} = 0.8L_{sys}$ ja profiilien arvot rajattiin maksimissaan kolmeen eri profiiliin. Vertailu Ketolan ja tämän työn tuloksiin on esitetty taulukossa 6.3.

Tässä työssä löydettiin lähes 90 kg kevyempi ristikko huomattavasti nopeammin. Ero ristikoiden massoissa johtuu lähes kokonaan paarteiden profiileista. Kuvassa 6.8 on esitetty ristikkorakenteen massan jakautuminen sauvojen kesken.

Taulukko 6.2. Diskreetit geometriaoptimoinnin tulokset

Paino (kg)	FEM-Analyysit	Aika (s)	Max con	
1479.44	3428	1194.4	0.0	I
1473.22	3351	1223.0	0.0	I
1464.47	1578	562.5	0.0	I
1437.12	745	276.0	0.0109	II
1506.93	1474	519.6	0.0	I
1427.22	598	200.3	0.0452	III
1488.76	1480	511.8	0.0	I
1576.44	1279	436.2	0.0035	II
1501.87	871	299.4	0.0416	I
1511.47	2688	914.1	0.0	IV
1486.69 ±42.20	1749.20 ±1041.84	613.72 ±370.56	-	

^I Vapaavälin minimiarvo

^{II} Yläpaarteen puristustaivutus

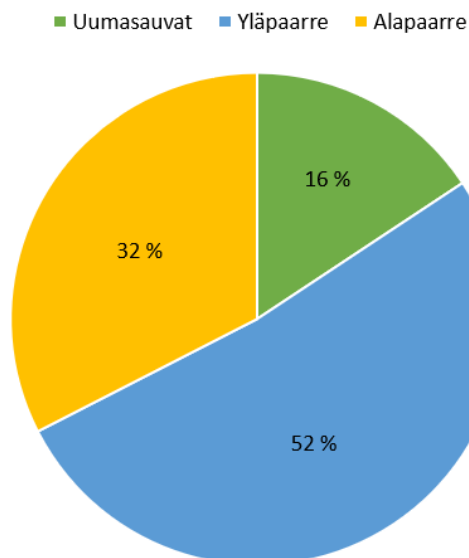
^{III} Paarteen pinnan myötääminen

^{IV} Katon kaltevuus

Taulukko 6.3. 24m ristikon geometriaoptimoinnin tulosten vertailu

Sauva	PSO + Mitoitus [22]	SLP + VNS
1	140x8.8	140x8
2	120x8	120x7.1
3	90x4	100x3
4	90x4	90x3
5	60x3	50x4
6	60x3	70x3
7	50x3	50x3
8	50x3	70x3
9	50x3	50x3
10	50x3	50x3
Paino (kg)	1556.96	1464.47
Kesto (min)	75-90	15-30

Paarteet muodostavat 84 % koko ristikon massasta, joten paarteiden valinnalla on todella suuri merkitys koko rakenteen kannalta. Ketolan [22] ja tämän työn optimituloksen paarteiden välinen ero on keskimäärin 3 kg / m ja paarteiden yhteenlaskettu pituus 24 metrin ristikolla on lähimain 40 metriä, josta tulee yhteensä 120 kg. Jäljelle jäävä 30 kg ero johtuu uumasauvojen profiileista sekä ristikon geometriasta.



Kuva 6.8. Ristikön massan jakautuminen eri sauvojen kesken

6.3.2 Uumasauvojen nurjahduspituuden vaikutus

Tarkastellaan uumasauvan nurjahduspituuden vaikutusta ristikon massaan. Vaikka kuvasta 6.8 nähdään, että uumasauvojen osuus koko ristikon massasta on pieni, voi nurjahduspituus vaikuttaa välillisesti optimigeometriaan, jonka kautta paarteille voidaan löytää pienemmät profiilit. Jatkuvan tehtävän tulokset on esitetty taulukossa 6.4.

Kaikissa tapauksissa optimointi päättyi SLP:llä 100 iteraatioon ja FEM-analyyseja tehtiin kussakin tapauksessa 3103. Tuloksissa ei juurikaan ole eroa niin poikkileikkausten dimensioiden, ristikon korkeuden kuin massankaan suhteen. Tehtävä diskretoidaan VNS-algoritmia hyödyntäen kuten edellisessä luvussa.

Parhaat diskreetit optimitulokset on esitetty taulukossa 6.5. Tulokset ovat hyvin lähellä toisiaan, joten tässä 24 metrin ristikon tapauksessa voidaan sanoa heuristisen algoritmin stokastisuudella olevan suurempi merkitys optimituloksen kannalta kuin uumasauvojen nurjahduspituuksilla.

6.3.3 36m ristikon geometriaoptimointi

Tarkastellaan 36 metrin ristikon optimointia 24 metrin ristikon optimoinnin huomioiden pohjalta. Tällä tavalla pystytään varmistamaan työssä esitettyjen menetelmien skaalautuvuudesta. Muuttujarajat sekä vapaavälin vakioarvo pidetään samana kuin 24 metrin tapauksessa.

Liitokset sijoitetaan paarteille tasavälein. Ristikön alkugeometria ja geometriarajat ovat

seuraavat

$$H_1 = 2400, H_2 = 3600, dx = 1800$$

$$1200 \leq H_1 \leq 3600$$

$$1800 \leq H_2 \leq 4400$$

(6.68)

Taulukko 6.4. Uumasauvan nurjahduspituuden vaikutus optimitulokseen

Muuttuja	$L_{cr} : 0.8$	$L_{cr} : 0.9$	$L_{cr} : 1.0$
h_1	141.42	139.97	141.83
t_1	6.76	6.88	6.64
h_2	122.78	125.53	125.74
t_2	6.58	6.54	6.57
h_3	105.99	105.93	105.49
t_3	3.13	3.124	3.10
h_4	85.49	90.71	90.73
t_4	3.19	3.0	3.00
h_5	57.85	59.015	60.36
t_5	3.40	3.22	3.11
h_6	71.61	69.94	73.15
t_6	3.00	3.00	3.00
h_7	51.21	50.00	50.50
t_7	3.00	3.00	3.00
h_8	70.17	69.12	73.66
t_8	3.00	3.00	3.00
h_9	52.65	50.82	50.00
t_9	3.00	3.00	3.00
h_{10}	50.00	50.00	50.00
t_{10}	3.00	3.00	3.00
H_1	1529.71	1551.46	1565.84
H_2	2729.71	2751.46	2765.84
Max con	0.079	0.061	0.052
Paino (kg)	1350.59	1363.98	1357.42

Taulukko 6.5. Nurjahduspituuden vaikutus diskreettiin tulokseen

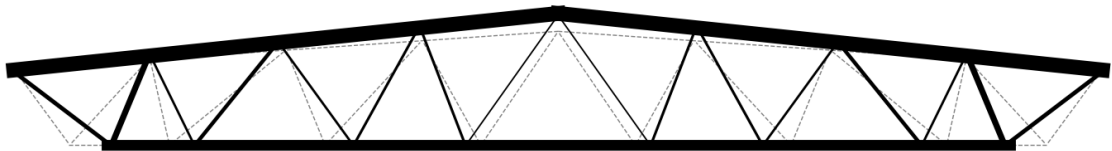
L_{cr}/L	Paino (kg)	FEM-Analyysit	Aika (s)	Max con	
0.8	1473.93	2793	982.5	0.0	I
0.9	1464.47	1578	562.5	0.0	I
1.0	1479.74	2840	954.7	-0.0	II

^I Vapaavälin minimiarvo

^{II} Yläpaarteen puristustaivutus

Optimoidun ristikon geometria on lähes identtinen 24 metrin ristikon kanssa. 36 metrin ristikon optimigeometria on esitetty kuvassa 6.9. Jatkuvan tehtävän diskretoinnissa ongelmaksi muodostui harjaa lähempänä olevien uumasauvojen liitosten geometria. Tämä huomataan diskreeteistä optimituloksista, jotka on esitetty taulukossa 6.6.

Mitä lähempänä uumasauva on harjaa, sitä pienempi normaalivoima siihen vaikuttaa eli voidaan käyttää pienempää profiilia. Paarteen profiili valikoituu kuitenkin koko paarteen matkalla vaikuttavien suurimpien rasitusten avulla. Uumasauvojen saadessa pienet profiilit ja paarteen suuren profiilin, leveyksien suhde β pienenee, jonka seurauksena vapaavälin minimiarvo kasvaa. Jos vapaaväliä kasvatetaan ilman että profiilit muuttuvat, epäkeskisyys kasvaa. Epäkeskisyyden kasvaessa sallittujen rajojen ulkopuolelle, on joko kasvatettava uumasauvojen kokoa tai muutettava liitoksen sijaintia. Uumasauvojen kasvattaminen vaikuttaa nyt myös uudelleen β :n arvoon, jolloin liitokselle sallittaisiinkin pienempi vapaavälin arvo. Tämä aiheuttaa optimointialgorimille todella paljon haasteita sillä uumasauvojen profiilin tai liitoksen sijainnin muuttaminen vaikuttaa myös uumasauvojen toisten päiden liitoksiin.



Kuva 6.9. 36 metrin ristikon optimigeometria verrattuna alkuperäiseen geometriaan

Taulukko 6.6. Diskreetit 36m geometriaoptimoinnin tulokset

Paino (kg)	FEM-Analyysit	Alka (s)	Max con	
3573.86	1279	398.4	0.0	I
3602.45	944	261.6	0.0	I
3569.36	1836	590.0	0.0	II
3586.79	1018	405.1	0.0	I
3566.65	1843	527.0	0.0	I
3572.15	1158	409.9	0.0	I
3554.53	2689	867.4	0.0	I
3610.97	972	260.3	0.0	I
3578.34	1013	300.0	0.0	I
3606.56	970	276.9	0.0	I
3582.16 ±18.89	1372.20 ±575.91	429.65 ±190.28	-	

I Vapaavälin minimiarvo

II Katon kaltevuus

36 metrin ristikon optimitulokset ja vertailu Ketolan [22] tuloksiin on esitetty taulukossa 6.7. Rakenteen massa jakautuu lähes samalla tavalla kuin 24 metrin ristikolla, 51 % yläpaarteelle, 31 % alapaarteelle ja loput 18 % massasta muodostaa uumasauvat. Vaikka Ketolan optimiratkaisussa alapaarre on pinta-alaltaan suurempi kuin tämän työn alapaar-

re, on hänen löytämät uumasauvojen profiilit paljon pienemmät. Kapeampi ja paksumpi alapaarre saattaa helpottaa liitosten geometriaehtojen toteutumista, joka voi johtaa pienempiin uumasauvojen profiileihin ja pienempään ristikon kokonaispainoon.

Taulukko 6.7. 36m ristikon geometriaoptimoinnin tulosten vertailu

Sauva	PSO + Mitoitus [22]	SLP	SLP + VNS
1	180x10	181.37x9.14	180x10
2	140x10	156.61x7.65	160x8
3	120x4	127.42x3.76	120x4
4	120x4	132.37x3.90	140x5
5	90x3	70.36x4.09	70x4
6	90x3	114.51x3.38	110x4
7	70x3	88.22x3.00	90x3
8	70x3	100.98x3.00	100x3
9	70x3	101.75x3.00	100x3
10	70x3	54.82x3.00	60x3
Paino (kg)	3461.87	3321.14	3554.53
Kesto (min)	80-120	10	15-30

Ristikko optimoitiin myös käyttäen 20 uumasauvaa. Jatkuvan tehtävän ratkaisuksi tuli 3598.74 kg, joka on hieman suurempi kuin 16 uumasauvalla optimoitu ristikko. Vaikka jatkuvan tehtävän ratkaisu löydettiin vaivatta, ei ongelmalle pystytty löytämään diskreettiä ratkaisua. Ongelmaksi muodostui liitosten geometriaehtojen täytyminen.

Kun sauvojen määrää kasvatetaan, uumasauvojen kulmat saavat keskimäärin suurempia arvoja. Tämän seurauksena epäkeskisyys kasvaa herkästi sallittujen rajojen ulkopuolelle. Uumasauvojen rasitukset jakautuvat myös tasaisemmin, joka johtaa keskimäärin pienempiin profiileihin ja vapaavälin pienin sallittu arvo kasvaa.

6.3.4 Huomioita ristikon optimoinnista

Useimmissa tapauksissa ristikon suurimmaksi rajoitusehdoksi tulee joko jokin liitoksen geometriaan liittyvä ehto, paarteen pinnan myötäminen tai yläpaarteen stabiilisuus yhdistetylle puristukselle ja taivutukselle. Työn tekemisen aikana testattiin myös jatkuvan tehtävän optimointi ilman liitosten rajoitusehtoja, mutta tämä johti kuitenkin tilanteeseen missä käypää diskreettiä ratkaisua ei pystytty enää löytämään.

Käytettäessä peräkkäisten linearisointien menetelmää jatkuvan tehtävän ratkaisemiseksi, on rajoitusehdot relaxoitava koko optimoinnin ajaksi. Jos rajoitusehtoja ei relaxoida, algoritmi päättyy hyvin nopeasti tilanteeseen missä se ei enää löydä käypiä ratkasuja. Tehtävä linearisoidaan yksi muuttuja kerrallaan, joka ei kuvasta rakenteen oikeaa käyttäytymistä kun muuttujia muutellaan samanaikaisesti, joten alitehtävän ratkaisuisissa päädy-

tään hyvin useasti kokonaistehtävän kannalta epäkäypään tulokseen tai alitehtävässä ei löydetä käypää ratkaisua.

Taulukko 6.8. 24m ristikon optimitulokset VNS:n avulla

Paino (kg)	FEM-Analyysit	Aika (s)	Max con	
2166.12	5723	1715.5	-0.014	I
2066.04	6391	1982.8	-0.004	I
2016.63	6108	1882.0	-0.003	I
2254.13	5184	1586.5	0.007	II
1976.51	6257	1931.5	0.0	III
1905.34	6321	1977.2	0.0	III
1803.11	6771	2092.7	-0.0	I
2145.89	7894	2436.0	-0.0	I
2036.13	6097	1801.2	0.0	III
1957.48	4882	1405.5	-0.003	I

I Yläpaarteen puristustaivutus

II Uumasauvan minimikulma

III Katon kaltevuus

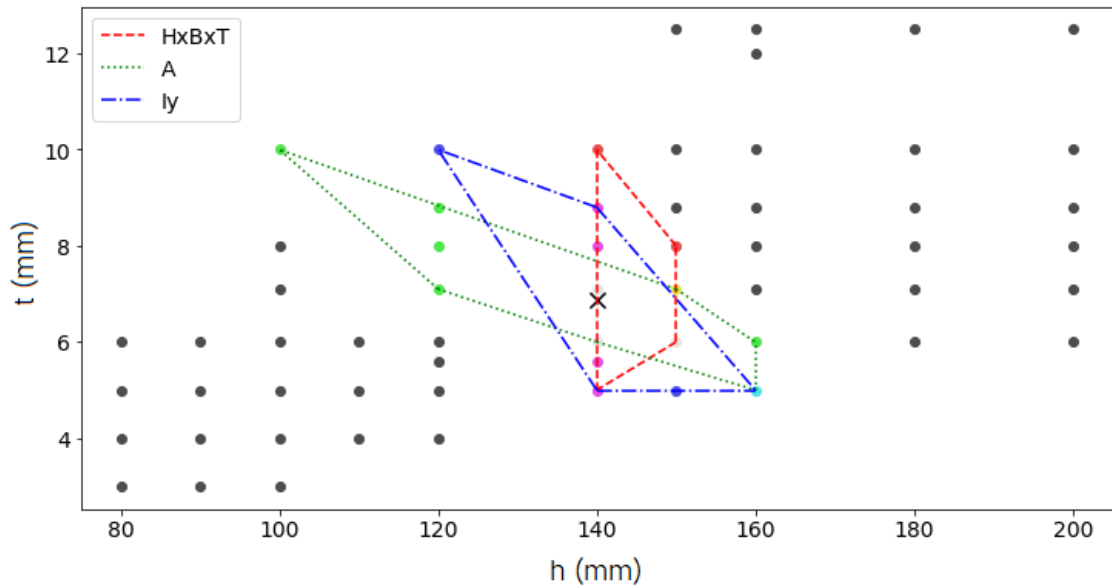
24 metrin ristikko optimoitiin myös suoraan VNS-algoritilla. Tulokset on esitetty taulukossa 6.8. Menetelmän hyvänä puolena voidaan pitää sitä, että kaikki löydettyt optimitulokset ovat valmiiksi diskreettejä ja myös käypiä. Huonona puolena on se, että tuloksen optimaalisuudesta ei ole takeita ja menetelmä on hyvin hidasku verrattuna jatkuvilla muuttujilla optimointiin. Iteraatioiden määrä rajattiin 100 iteraatioon.

Optimointia testattiin myös rajoittamatta VNS:n iteraatioiden lukumäärää. Tästä huolimatta algoritmi ei löytänyt 1800 kg keveyempää ristikkoa ja optimointi kesti lähes 52 minuuttia. Vaikka VNS saattaisi löytää paremman tuloksen useammalla yrityksellä, ei sitä voida pitää tällä implementoinnilla hyvänä globaalina optimointimenetelmänä eurokoodin mukaista ristikkoa optimoitaessa.

Jatkuvan ratkaisun ympäriltä luotavien alilistojen tarkoitus on rajata diskreetin tehtävän hakuvaruutta. Käytettävät parametrit lähimpien profiilien valinnassa vaikuttaa diskreetissä tehtävässä käytössä oleviin profiileihin. Kuvassa 6.10 on esitetty 10 lähimmän diskreetin profiilin valinta dimensioiden, pinta-alan ja neliömomentin suhteen. Kuvassa piste X on 24 metrin ristikon jatkuvan tehtävän yläpaarteen poikkileikkauksen dimensiot. Ilman tarkempaa tutkimista on vaikea sanoa mitkä ovat parhaat parametrit alilistan luomiseen.

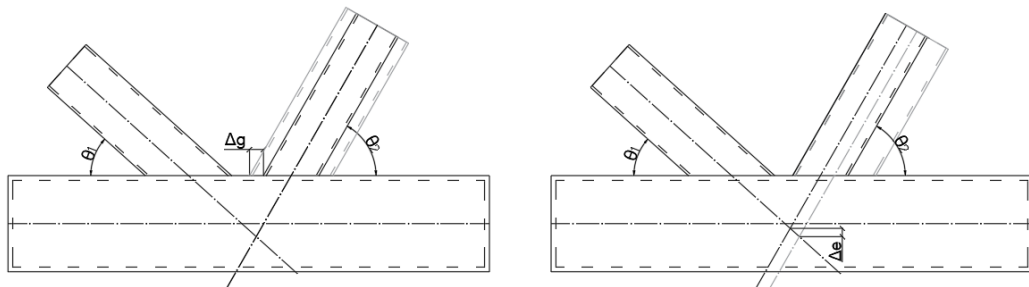
6.3.5 Liitosmitoituksen haasteet

Liitosten yhtäaikainen mitoitus muun optimoinnin kanssa lisää huomattavasti haastetta löytää käypiä ratkaisuja. Tehtävää muodostettaessa on päätettävä miten liitoksen geometria muuttuu sauvoja, kulmia ja vapaaväliä tai epäkeskisyyttä muutettaessa.



Kuva 6.10. Eri parametreilla indeksimuuttujan alilista muodostuu eri profiileista

Geometriaoptimoinnissa testattiin myös vapaavälin vakioarvona 20 mm ja 30 mm. Kummassakin tapauksessa jatkuvan tehtävän tulokset näyttivät hyvältä, mutta diskretoinnissa käypiä ratkaisuja löytyi todella harvoin, jos ollenkaan. Syynä tähän on luultavasti profiilien koon kasvaminen siirryttäessä jatkuvista diskreetteihin poikkileikkauksiin. Suurella vapaavälin vakioarvolla diskreetille tehtävälle jää enemmän pelivaraa kasvattaa tai pienentää vapaavälin arvoa sekä liitoksen sauvoja, pitäen epäkeskisyyden kuitenkin sallituissa rajoissa.



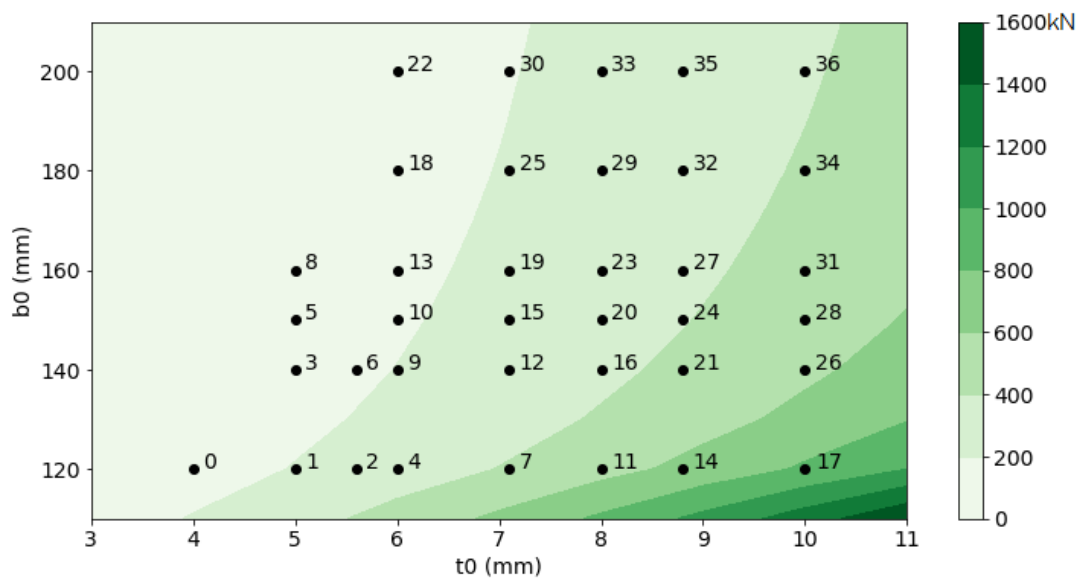
Kuva 6.11. Uumasauvan profiilin muutos vaikuttaa liitoksen vapaaväliin ja epäkeskisyyteen

Tarkastellaan tilannetta, jossa K-liitoksen toista uumasauvaa kasvatetaan. Jos sauvan keskilinja pidetään paikallaan, vapaaväli pienenee sauvan molempien päiden liitoksissa ja vapaavälin rajoitusehdot tiukentuvat, sillä β kasvaa. Pidettäessä vapaaväli vakiona, uumasauvan molempien päiden liitosten keskipiste siirtyy paarteen akselilla ja epäkeskisyydet kasvavat. Uumasauvan kasvattaminen saattaa siis johtaa tilanteeseen, jossa liitoksen geometriarajat rikkoutuvat ja ratkaisusta tulee epäkäypä. Kuvassa 6.11 on havainnollistettu miten liitoksen epäkeskisyydet ja vapaaväli muuttuu kun uumasauvan kokoa muutetaan. Vasemman puoleisessa kuvassa liitoksen epäkeskisyydet pidetään vakiona ja

oikean puoleisessa vapaaväli.

Geometrian optimoinnissa liitosten vapaavälit ja epäkeskisyydet järjestyvät rajoitusehtojen mukaan, kun solmuja liikutellaan vapaasti samalla, kun profiileja vaihdellaan. Solmujen sijainnit voidaan ilmoittaa erilaisilla muuttujavalinnoilla, mutta lopulta muuttujina ovat uumasauvojen keskilinjojen risteyskohdat.

Tässä työssä ristikot optimointiin pitämällä vapaavälin arvo muuttumattomana sauvojen profiilien muuttuessa. Toinen vaihtoehto olisi pitää liitoksen epäkeskisyyden vakiona. Tämä voi kuitenkin johtaa tilanteeseen, missä sauvojen profiilien muuttuessa vapaaväli häviää kokonaan pois tai muuttuu limitykseksi. Liitoksen kestävyyslaskennasta saadaan tällöin virheellisiä tuloksia, joka voi johtaa optimointialgoritmin epäloogiseen käyttäytymiseen ja optimoinnin keskeytymiseen.



Kuva 6.12. Paarteen seinämäpaksuuden ja leveyden vaikutus paarteen pinnan kestävyyteen

Käytettäessä profiilikatalogin mukaisia poikkileikkauksia, on kiinnitettävä erityistä huomiota profiilien järjestykseen. Tarkastellaan ristikon tuella olevaa vedetyn uumasauvan ja paarteen välistä Y-liitosta. Kuvassa 6.12 on esitetty paarteen pinnan murtokestävyys paarteen leveyden ja seinämäpaksuuden avulla. Uumasauvan leveys on 100 mm ja se liittyy paarteeseen 45° kulmassa. Paarteen jännitykselle on käytetty arvoa $n = 0.5$. Kuvassa esitetyt pisteet ja numerot ovat diskreettejä poikkileikkauksia sekä niiden järjestysnumero pinta-alan mukaan järjestetyssä profiilikatalogissa. Kuvasta nähdään, että paarteen kasvattaminen pinta-alan mukaan järjestetyssä katalogissa saattaa siis johtaa tilanteeseen missä paarteen pinta myötää uumasauvan vetorasituksesta. Katalogin järjestäminen liitoksen kestävyyslaskennan avulla ratkaisisi tämän esimerkin mukaisen ongelman, mutta sama ongelma saattaisi esiintyä nyt muualla, esimerkiksi paarteen nurjahduskestävyydessä. Ristikon geometrian optimoinnissa kaikkien sauvojen profiilit sekä liitosten sijainnit saattavat muuttua samanaikaisesti, kestävyyslaskennat määrättyvät epälineaarisesti ja rasiitukset saadaan epäsuorasti FEM-analyysin avulla mikä tekee tehtävästä todella haastavan.

7 YHTEENVETO

Tulevaisuuden haasteet kuten ilmastonmuutos sekä uunikit rakennuskohteet ajavat suunnittelijat tilanteeseen, missä suunnitteluratkaisussa pitää kiinnittää huomiota toimivuuden lisäksi myös resurssitehokkuuteen. Suuri osa teräsrakenteista muodostuu standardiprofiileista, joka tekee suunnittelutehtävästä diskreetin. Diskreettien optimointitehtävien haasteena on hakuavaruuden laajuus, sillä tehtävä on eräänlainen kombinatorinen tehtävä missä hakuavaruus kasvaa eksponentiaalisesti käytettävissä olevien profiilien mukaan. Optimointi on työkalu, jolla rakenteiden suunnittelua pystytään automatisoimaan ja se tarjoaa suunnittelijalle mahdollisuuden vertailla erilaisia rakenneratkaisuja vaivatta. Kaikki tehtävät, jotka voidaan kirjoittaa optimointitehtävän muotoon pystytään ratkaisemaan.

Tehtävän muotoilu ja suunnittelumuuttujien valinta vaikuttaa käytettävissä oleviin optimointialgoritmeihin ja sitä kautta lopputulokseen. Kaikkia optimointitehtäviä ei pystytä ratkaisemaan samalla menetelmällä, vaan optimointitehtävä on muodostettava tapauskohtaisesti miettien tehtäväkohtaisia haasteita. Jos päädytään tilanteeseen, missä käypää tulosta on vaikea löytää, on tehtävää helpotettava ja pilkottava pienempiin osiin.

Yksi tapa lähestyä diskreettiä optimointiongelmaa on relaksoida käytössä olevat diskreetit arvot ja ratkaista tehtävä kahdessa vaiheessa. Ensimmäisessä vaiheessa tehtävä ratkaistaan jatkuvien muuttujien avulla kuvaamalla alkuperäistä diskreettiä arvoa joko yhdellä tai useammalla muuttujalla. Muuttujien arvot on rajattava diskreettien arvojen mukaisesti, sillä muuten jatkuvan ratkaisun ympäriltä on hyvin vaikea löytää käypää diskreettiä tulosta. Toisessa vaiheessa jatkuvan tehtävän ratkaisu diskretoidaan esimerkiksi haaroittamalla tehtävä yksi muuttuja kerrallaan, kunnes päädytään diskreettiin tulokseen.

Yksinkertaisin ja kirjallisuudessa yleisesti käytetty menetelmä diskreettien tehtävien ratkaisemiseksi on käyttää indeksimuuttujia, eli diskreettien arvojen järjestysnumeroa suunnittelumuuttujana. Tällä tavalla poikkileikkauksen ominaisuudet pystytään lukemaan suoraan taulukosta optimoinnin aikana. Indeksimuuttujien kanssa esitetty optimointitehtävä ratkaistaan yleensä käyttäen jotain metaheuristista algoritmia, kuten geneettistä algoritmia.

Optimointialgoritmeja on monia, mutta ne voidaan pääasiassa jakaa kahteen ryhmään: deterministisiin ja metaheuristisiin algoritmeihin. Käytettäessä deterministisiä algoritmeja on tehtävä pystyttävä relaksoimaan, jotta gradienttietoa pystytään hyödyntämään. Tapauksissa, jossa tehtävä ei ole relaksoitavissa on käytettävä heuristisia menetelmiä. Niiden hyötynä on joustavuus, sillä ne soveltuvat lähes kaikentyypisten optimointiongel-

mien ratkaisemiseen. Heurististen menetelmien huonona puolena voidaan pitää niiden stokastisuutta, eli satunnaisuutta, joka vaikuttaa huomattavasti optimituloksen löytämiseen ja tekee saman ratkaisun löytämisestä lähes mahdotonta.

Kirjallisuudessa yleisesti käytetyt rakenteet optimoitiin käyttämällä jatkuvia-, diskreetti- sekä indeksimuuttujia. Diskreetti optimointitehtävä relaxsoitiin ja ratkaistiin käyttäen perättäisten linearisointien menetelmää (SLP), missä tehtävä linearisoidaan joka iteraatiopisteessä jokaisen suunnittelumuuttujan suhteen. Jatkuvan ratkaisun ympäriltä valittiin lähimmät diskreetit arvot euklidisen etäisyyden perusteella ja tehtävä diskretoitiin indeksimuuttujien avulla käyttäen muuttuvan ympäristön hakua (VNS). Lokaali haku suoritettiin geneettisen algoritmin (GA) avulla.

Osa tehtävistä ratkaistiin suoraan käyttäen diskreettejä muuttujia ja peräkkäistä sekaluoketehtävän linearisointialgoritmia (MISLP). Diskreettien muuttujien avulla lausuttu tehtävä linearisoidaan ja linearisoitu alitehtävä ratkaistaan käyttäen binäärimuuttujia, joiden avulla alitehtävän ratkaisuna saadaan diskreetti ratkaisu. Kaikkia tehtäviä ei saatu ratkaistua tällä menetelmällä vaan osassa tehtävistä algoritmi jäi vaihtelemaan kahden tai useamman epäkäyvän pisteen välille.

Luvussa 5 esitellyt tehtävät ovat hyviä erilaisten menetelmien nopeaan toimivuuden testaamiseen, mutta ovat kuitenkin hyvin kaukana todellisuuden suunnittelutehtävistä. Eurokoodin mukaisen ristikon optimoinnista huomataan, että suurin osa ajasta kuluu itse tehtävän muodostamiseen eikä sen ratkaisemiseen. Geometriaoptimointi ja liitoksen mitoitus samanaikaisesti tekee tehtävästä todella haastavan sillä pienikin muutos poikkileikkauksen mitoissa tai liitoksen sijainnissa voi johtaa epäkäypään ratkaisuun. Vastoin perinteistä käsitystä, sauvan profiilin kasvattaminen saattaa johtaa epäkäypään tulokseen kuten kuvasta 6.12 huomataan. Jatkotutkimuksena voitaisiin tutkia ylä- ja alapaarteen päätyliitosten vahvistamisen vaikutusta optimitulokseen. Tällöin kohdefunktiona tulisi käyttää massan sijasta valmistuskustannuksia.

Ristikon paarteet muodostavat suurimman osan ristikon massasta. Hyvänä nyrkkisääntönä voidaan pitää, että ristikon paino jakautuu ylä- ja alapaarteen sekä uumasauvojen kesken 50-30-20 -suhteessa. Geometriaoptimoinnin tarkoituksena ei ole minimoida uumasauvojen nurjahduspituuksia, vaan löytää liitoksille sellaiset sijainnit, missä yläpaarteen rasitukset jakautuvat mahdollisimman tasaisesti jokaiselle solmuvälille. Ristikon taipuma pystytään myös minimoimaan geometriaoptimoinnin avulla.

Kattokaltevuus on rajoitettava ristikon geometriaoptimoinnissa, sillä harjakorkeuden kasvaessa uumasauvojen liitoskulmat kasvavat, jonka seurauksena liitoksen geometrialle on vaikea löytää käyviä ratkaisuja. Toisena vaihtoehtona olisi rajoittaa liittyvien uumasauvojen kulmien summaa.

Suuri osa työn tekemiseen kuluva ajasta meni optimointiympäristön kehittämiseen. Kristo Melan kanssa yhteistyössä kehitetty optimointikehys mahdollistaa myös muiden rakenteiden kuin teräsristikoiden optimoinnin. Optimointiongelmia on helppo luoda ja tehtävän ratkaisemiseen käytettävät suunnittelumuuttujat on helposti valittavissa ja niiden

tyyppi on vaihdettavissa tarpeen vaatiessa. Ohjelmaan implementoidut ratkaisijat, eli optimointialgoritmit, on pyritty toteuttamaan siten, että algoritmi on helposti vaihdettavissa jos alkuperäinen lähestymistapa ei tuota haluttua tulosta.

LÄHTEET

- [1] J. S. Arora. *Introduction to optimum design*. Elsevier, 2004.
- [2] T. Augusto, K. Mounir ja A. M. Melo. A cost optimization-based design of precast concrete floors using genetic algorithms. *Automation in construction* 22 (2012), 348–356.
- [3] J. Cai ja G. THIERAUF. Discrete optimization of structures using an improved penalty function method. *Engineering Optimization* 21.4 (1993), 293–306.
- [4] E. S. Croce, E. G. Ferreira, A. C. Lemonge, L. G. Fonseca ja H. J. Barbosa. A genetic algorithm for structural optimization of steel truss roofs. *XXV CILAMCE, 25th Iberian Latin-American Congress on Computational Methods in Engineering, UFPE, Recife, PE, Brasil*. 2004.
- [5] D. T. Do ja J. Lee. A modified symbiotic organisms search (mSOS) algorithm for optimization of pin-jointed structures. *Applied soft computing* 61 (2017), 683–699.
- [6] *Eurocode 3: Design of steel structures. Part 1-1: General rules and rules for buildings*. EN 1993-1-1:2005. Helsinki, Finland: Technology Industries of Finland, Standards, 2005.
- [7] *Eurocode 3: Design of steel structures. Part 1-8: Design of joints*. EN 1993-1-8:2005. Helsinki, Finland: Technology Industries of Finland, Standards, 2005.
- [8] J. Forrest, T. Ralphs, S. Vigerske, LouHafer, B. Kristjansson, jpfasano, EdwinStraver, M. Lubin, H. G. Santos, rlougee ja M. Saltzman. *coin-or/Cbc: Version 2.9.9*. Versio releases/2.9.9. Heinäkuu 2018. DOI: 10.5281/zenodo.1317566. URL: <https://doi.org/10.5281/zenodo.1317566>.
- [9] M. Gendreau ja J.-Y. Potvin. Metaheuristics in combinatorial optimization. *Annals of Operations Research* 140.1 (2005), 189–213.
- [10] A. Groenwold, N. Stander ja J. Snyman. A regional genetic algorithm for the discrete optimal design of truss structures. *International Journal for Numerical Methods in Engineering* 44.6 (1999), 749–766.
- [11] Y. Guerin. How to Deal with Mixed-Variable Optimization Problems: An Overview of Algorithms and Formulations. *Advances in Structural and Multidisciplinary Optimization: Proceedings of the 12th World Congress of Structural and Multidisciplinary Optimization (WCSMO12)*. Springer. 2017, 64.
- [12] R. T. Haftka ja Z. Gürdal. *Elements of structural optimization*. Vol. 11. Springer Science & Business Media, 2012.
- [13] K. Hager ja R. Balling. New approach for discrete structural optimization. *Journal of Structural Engineering* 114.5 (1988), 1120–1134.
- [14] P. Hansen ja N. Mladenović. Variable neighborhood search: Principles and applications. *European journal of operational research* 130.3 (2001), 449–467.

- [15] P. Hansen, N. Mladenović, R. Todosijević ja S. Hanafi. Variable neighborhood search: basics and variants. *EURO Journal on Computational Optimization* 5.3 (2017), 423–454.
- [16] O. Hasançebi ja S. K. Azad. An exponential big bang-big crunch algorithm for discrete design optimization of steel frames. *Computers & Structures* 110 (2012), 167–179.
- [17] O. Hasançebi ja S. K. Azad. Improving Computational Efficiency of Bat-Inspired Algorithm in Optimal Structural Design. *Advances in Structural Engineering* 18.7 (2015), 1003–1015.
- [18] M.-W. Huang ja J. S. Arora. Optimal design of steel structures using standard sections. *Structural optimization* 14.1 (1997), 24–35.
- [19] J. Jalkanen. *Tubular truss optimization using heuristic algorithms*. Tampere University of Technology, 2007.
- [20] D. Juang ja W. Chang. A revised discrete Lagrangian-based search algorithm for the optimal design of skeletal structures using available sections. *Structural and Multidisciplinary optimization* 31.3 (2006), 201–210.
- [21] A. Kaveh ja A. Zolghadr. A multi-set charged system search for truss optimization with variables of different natures; element grouping. *Periodica Polytechnica Civil Engineering* 55.2 (2011), 87–98.
- [22] T. Ketola. Teräsristikon algoritmiavusteinen suunnittelu ja optimointi. Tutkielma. 2019.
- [23] S. Kirkpatrick ja G. Toulouse. Configuration space analysis of travelling salesman problems. *Journal de Physique* 46.8 (1985), 1277–1292.
- [24] U. Kirsch. *Structural optimization: fundamentals and applications*. Springer Science & Business Media, 1993.
- [25] S. Koziel ja X.-S. Yang. *Computational optimization, methods and algorithms*. Vol. 356. Springer, 2011.
- [26] S. Kravanja, T. Žula ja U. Klanšek. Discrete optimization of steel frame structures. *Proceedings of the 2nd WSEAS International Conference on Computer Engineering and Applications*. World Scientific, Engineering Academy ja Society (WSEAS). 2008, 131–136.
- [27] L. Li, Z. Huang ja F. Liu. A heuristic particle swarm optimization method for truss structures with discrete variables. *Computers & Structures* 87.7-8 (2009), 435–443.
- [28] Y. Li, S. Wang ja M. Han. Truss Structure Optimization Based on Improved Chicken Swarm Optimization Algorithm. *Advances in Civil Engineering* 2019 (2019).
- [29] M.-H. Lin, J.-F. Tsai ja C.-S. Yu. A review of deterministic optimization methods in engineering and management. *Mathematical Problems in Engineering* 2012 (2012).
- [30] H. Rahami, A. Kaveh ja Y. Gholipour. Sizing, geometry and topology optimization of trusses via force method and genetic algorithm. *Engineering Structures* 30.9 (2008), 2360–2369.
- [31] M. Safe, J. Carballido, I. Ponzoni ja N. Brignole. On stopping criteria for genetic algorithms. *Brazilian Symposium on Artificial Intelligence*. Springer. 2004, 405–413.

- [32] S. Shojaee, M. Arjomand ja M. Khatibinia. A hybrid algorithm for sizing and layout optimization of truss structures combining discrete PSO and convex approximation. *International Journal of Optimization in Civil Engineering* 3.1 (2013), 57–83.
- [33] A. R. Toakley. Optimum design using available sections. *Journal of the Structural Division* 94.5 (1968), 1219–1244.
- [34] M. R. Vezvari, A. Ghoddosian ja A. Nikoobin. Numbers Cup Optimization: A new method for optimization problems. *Structural Engineering and Mechanics* 66.4 (2018), 465–476.
- [35] X. Yu, E. H. Johnson ja S. Zhang. Discrete Optimization in MSC. Nastran. *MSC World User's Conference*. 2000.

A OPTIMOINNISSA KÄYTETYT DISKREETIT POIKKILEIKKAUKSET

Taulukko A.1. 10-sauvan optimoinnissa käytettävät pinta-alat, A_{10}

1044.9 ,	1161.0 ,	1283.55 ,	1373.85 ,	1535.1 ,
1689.9 ,	1696.35 ,	1857.6 ,	1889.85 ,	1993.05 ,
2018.85 ,	2180.1 ,	2238.15 ,	2289.75 ,	2341.35 ,
2476.8 ,	2496.15 ,	2502.6 ,	2696.1 ,	2721.9 ,
2896.05 ,	2960.55 ,	3096.0 ,	3205.65 ,	3302.4 ,
3702.3 ,	4656.9 ,	5140.65 ,	7417.5 ,	8707.5 ,
8965.5 ,	9159.0 ,	9997.5 ,	10320.0 ,	10900.5 ,
12126.0 ,	12835.5 ,	14190.0 ,	14770.5 ,	17092.5 ,
19350.0 ,	21607.5			

Taulukko A.2. 15-sauvan optimoinnissa käytettävät pinta-alat, A_{15}

71.613 ,	90.968 ,	112.258 ,	141.935 ,	174.193 ,
185.161 ,	223.871 ,	283.87 ,	347.741 ,	615.483 ,
697.418 ,	757.418 ,	859.998 ,	959.998 ,	1138.062 ,
1381.933 ,	1739.997 ,	1806.448 ,	2019.996 ,	2299.995 ,
2459.995 ,	3099.994 ,	3839.992 ,	4239.992 ,	4639.991 ,
5499.989 ,	5999.988 ,	6999.986 ,	8580.628 ,	9219.336 ,
11077.397 ,	12374.169			

Taulukko A.3. 52-sauvan optimoinnissa käytettävät pinta-alat, A_{52}

71.613 ,	90.968 ,	126.451 ,	161.29 ,	198.064 ,
252.258 ,	285.161 ,	363.225 ,	388.386 ,	494.193 ,
506.451 ,	641.289 ,	645.16 ,	729.031 ,	792.256 ,
816.773 ,	939.998 ,	1008.385 ,	1045.159 ,	1161.288 ,
1283.868 ,	1374.191 ,	1535.481 ,	1690.319 ,	1696.771 ,
1858.061 ,	1890.319 ,	1993.544 ,	2180.641 ,	2238.705 ,
2290.318 ,	2341.931 ,	2477.717 ,	2496.769 ,	2503.221 ,
2696.769 ,	2722.575 ,	2896.768 ,	2961.284 ,	3096.768 ,
3206.445 ,	3303.219 ,	3703.218 ,	4658.055 ,	5141.925 ,
5503.215 ,	5999.988 ,	6999.986 ,	7419.43 ,	8709.66 ,
8967.724 ,	9161.272 ,	9999.98 ,	10322.56 ,	10903.2 ,
12129.01 ,	12838.68 ,	14193.52 ,	14774.16 ,	15806.42 ,
17096.74 ,	18064.48 ,	19354.8 ,	21612.86	

Taulukko A.4. Ristikon optimoinnissa käytetty SHS-profiilikatalogi

h [mm]	t [mm]	h [mm]	t [mm]	h [mm]	t [mm]
50.0	3.0	120.0	5.6	180.0	6.0
50.0	4.0	120.0	6.0	180.0	7.1
50.0	5.0	120.0	7.1	180.0	8.0
60.0	3.0	120.0	8.0	180.0	8.8
60.0	4.0	120.0	8.8	180.0	10.0
60.0	5.0	120.0	10.0	180.0	12.5
70.0	3.0	140.0	5.0	200.0	6.0
70.0	4.0	140.0	5.6	200.0	7.1
70.0	5.0	140.0	6.0	200.0	8.0
80.0	3.0	140.0	7.1	200.0	8.8
80.0	4.0	140.0	8.0	200.0	10.0
80.0	5.0	140.0	8.8	200.0	12.5
80.0	6.0	140.0	10.0	220.0	7.1
90.0	3.0	150.0	5.0	220.0	8.0
90.0	4.0	150.0	6.0	220.0	8.8
90.0	5.0	150.0	7.1	220.0	10.0
90.0	6.0	150.0	8.0	220.0	12.5
100.0	3.0	150.0	8.8	250.0	8.0
100.0	4.0	150.0	10.0	250.0	8.8
100.0	5.0	150.0	12.5	250.0	10.0
100.0	6.0	160.0	5.0	250.0	12.5
100.0	7.1	160.0	6.0	260.0	8.0
100.0	8.0	160.0	7.1	260.0	8.8
100.0	10.0	160.0	8.0	260.0	10.0
110.0	4.0	160.0	8.8	260.0	11.0
110.0	5.0	160.0	10.0	260.0	12.5
110.0	6.0	160.0	12.0	300.0	10.0
120.0	4.0	160.0	12.5	300.0	12.5
120.0	5.0				