Tapio Mylläri

Keinotekoisten neuroverkkojen käyttö kombinatoristen optimointiongelmien ratkaisemisessa

Tietotekniikan kandidaatintutkielma

2. marraskuuta 2018

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Tapio Mylläri

Yhteystiedot: `tajomyll@student.jyu.fi`

Työn nimi: Keinotekoisten neuroverkkojen käyttö kombinatoristen optimointiongelmien ratkaisemisessa

Title in English: The usage of artificial neural networks for solving combinatorial optimization problems.

Työ: Kandidaatintutkielma

Sivumäärä: 19+0

Tiivistelmä: Keinotekoisia neuroverkkoja voidaan käyttää monenlaisten haastavien ongelmien ratkaisemiseen. Tällaisia ongelmia ovat esimerkiksi kuvantunnistus, äänentunnistus ja tekoälysovellukset monimutkaisempiin peleihin. Tällaisten toistuvien hahmojen ja muotojen tunnistamisen lisäksi keinotekoisia neuroverkkoja voidaan käyttää laskentaan, ja siten esimerkiksi optimointiin. Tässä kirjallisuuskatsauksessa selvitetään neuroverkkojen mahdollista käyttöä NP-vaikeiden kombinatoristen optimointiongelmien ratkaisemisessa ja perehdytään aiheesta tehtyyn tutkimukseen.

Avainsanat: neuroverkot, optimointi, NP-vaikeus, algoritmit

Abstract: Artificial neural networks can be used to solve a variety of difficult problems. For example image and voice recognition, and implementing artificial intelligence in complicated systems are very difficult problems to solve with traditional algorithms. More than that, artificial neural networks can be used to do regular mathematical calculations, and they can be used for optimization problems as well. The goal of this literature review is to find out how effective neural networks are for solving NP-hard combinatorial optimization problems.

Keywords: neural networks, optimization, NP-hard, algorithms

Kuviot

Sisältö

1 Johdanto

Keinotekoisia neuroverkkoja käytetään nykyään monenlaisiin eri tekoälysovelluksiin. Deepart-nimisellä verkkosivulla voi yhdistää esimerkiksi valokuvan ja jonkin taideteoksen tyyliä keskenään (DeepArt 2018). Googlen omistaman DeepMind-yrityksen AlphaGo Zero oppii pelaamaan Go:ta pelaamalla itseään vastaan (Silver ym. 2017). Näissä sovelluksissa on hyödynnetty neuroverkkoja ja syvää oppimista ongelmien ratkaisemiseen hyvällä menestyksellä. Koska neuroverkot voivat tunnistaa toistuvia elementtejä ja niiden kautta ratkaista vastaavia ongelmia, olisiko mahdollista käyttää niitä heuristisesti NP-vaikeiden ongelmien approksimoimisessa?

Tämän kandidaatintutkielman tavoitteena on selvittää keinotekoisten neuroverkkojen käyttökelpoisuus NP-vaikeiden optimointiongelmien ratkaisemisessa. Tarkastelu on rajattu kahteen klassiseen kombinatoriseen optimointiongelmaan: kauppamatkustajan ongelmaan ja kapsäkkiongelmaan ja näiden variaatioihin. Villarubia ym. (2018) kirjoittamassa artikkelissa on selvitetty keinotekoisten neuroverkkojen käyttöä yleisten lineaaristen ja epälineaaristen optimointitehtävien ratkaisemiseen hyvillä tuloksilla. Tässä kandidaatintutkielmassa tarkastelun kohteena on samalla selvittää keinotekoisiin neuroverkkoihin perustuvien menetelmien tehokkuutta muihin tunnettuihin heuristisiin menetelmiin nähden.

Seuraavassa luvussa käsitellään neuroverkkojen taustalla olevaa teoriaa ja historiallista taustaa tarkkoihin yksityiskohtiin menemättä. Sen jälkeen tarkastellaan käsiteltävien optimointitehtävien teoreettista taustaa ja aikaisempia heuristisia ratkaisukeinoja. Neljännessä luvussa käydään läpi erilaisia tutkimustuloksia ja tarkastellaan neuroverkkojen käytön hyötyä kyseisten NP-vaikeiden optimointiongelmien approksimoimisessa. Viimeisessä luvussa luodaan yhteenveto kaikesta siihen mennessä käsitellystä aiheesta.

2 Keinotekoiset neuroverkot

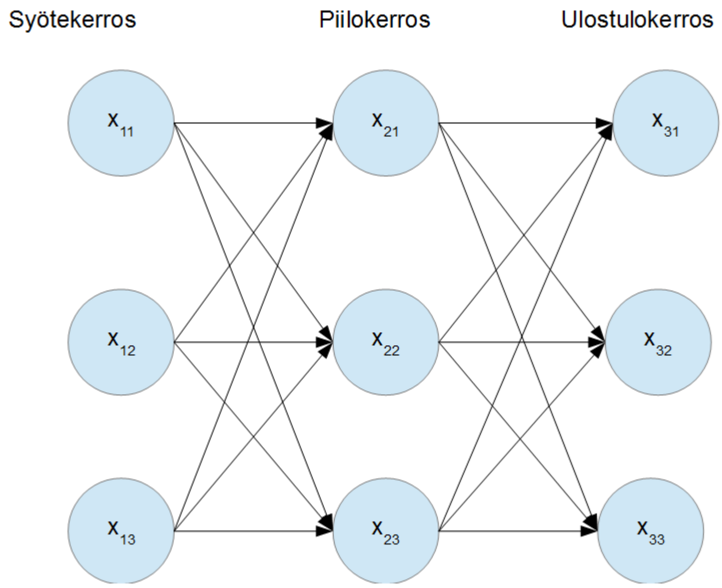
Keinotekoisten neuroverkkojen (engl. artificial neural network, ANN) idea perustuu aivoissa olevien biologisten neuronien toimintaan. Jo vuonna 1943 McCullochin ja Pittsin kirjoittamassa artikkelissa muodostettiin matemaattista todistusta neuronien muodostaman verkon mahdollisuudelle kuvata mitä tahansa loogisia lausekkeita. Vuonna 1958 Rosenblattin artikkelissa määriteltiin perseptronin, eli yksinkertaisimman keinotekoisen neuronin rakenne, minkä jälkeen eri tutkijat ovat vieneet neuroverkkojen tutkimusta eteenpäin.

Tyypilliseen ANN:ään kuuluu kolme kerrosta: syötekerros, piilokerros ja ulostulokerros. Jokainen tällainen kerros koostuu neuroneista ja kaikki kerroksen neuronit kytkeytyvät seuraavan kerroksen kaikkiin neuroneihin. Kuviossa 1 on kuvattu tällaisen neuroverkon rakennetta. Syötekerros on ensimmäinen kerros, jonka neuroneihin kuvataan jokin haluttu syöte. Esimerkiksi kuvan kaikki pikselit voidaan kuvata syötekerroksen neuronien avulla. Viimeisellä kerroksella on ulostulokerros, jonka neuronit vastaavat tietynmuotoista haluttua vastausta ongelmaan, eli neuroverkon vastaus käy ilmi ulostulokerroksen neuronien arvoista. Sellaista verkkoa, jossa aktivoinnit kulkevat yhteen suuntaan kutsutaan myötäkytketyksi(engl. feed forward) verkoksi. Näiden kahden kerroksen välinen alue on piilokerros, joka voi koostua useammasta erillisestä kerroksesta. Kyseisissä kerroksissa voi olla mielivaltainen määrä neuroneita.

Neuronien välinen aktivointi tapahtuu aktivointifunktion ja painotetun summan avulla. Kaikkien neuronien painot ja vakiotermit voidaan laskea yhteen, jolloin saadaan painotettu summa

$$\sum_{i=1}^{n} w_{ij}x_i + b_j, \tag{2.1}$$

missä w_{ij} on neuronin x_i ja x_j välinen kytkentä, x_i, $i = 1,2,...,n$ on edellisen kerroksen neuroni ja $b_j \in \mathbb{R}$ on seuraavan kerroksen neuroniin liittyvä vakiotermi (engl. bias). Nyt seuraavan kerroksen neuronin x_j, $j = 1,2,...,m$ arvo saadaan sijoittamalla tämä painotettu summa aktivointifunktioon. Aktivointifunktioksi kelpaisi teoriassa mikä



Kuvio 1. Esimerkkikuva tyypillisen eteenpäinkytketyn ANN:n rakenteesta.

tahansa funktio, mutta koska myöhemmin halutaan neuroverkon antamien arvojen konvergoivan luotettavasti, niin yleensä tämä painotettu summa kuvataan nollan ja ykkösen välille. Tyyppiesimerkki tällaisesta funktiosta on sigmoid-funktio, joka kuvaa annetun luvun nollan ja ykkösen välille. Tällainen funktio on esimerkiksi

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.2}$$

Seuraavan kerroksen neuronille asetetaan se tulos, mikä saadaan kun painotettu summa sijoitetaan sigmoid-funktioon. Tätä jatketaan kunnes lopulta ulostulokerroksen neuronit aktivoidaan ja neuroverkko on päässyt lopputulokseen, minkä jälkeen tulosta voidaan verrata oikeaan ratkaisuun ja sen perusteella muuttaa painoja ja vakiotermejä, eli niin sanotusti opettaa neuroverkkoa.

Neuroverkoilla harvoin saadaan tarpeeksi tarkkoja tuloksia ensimmäisellä laskentakerralla. Hyvin usein verkon vastaukset ovat käytännössä satunnaisesti annettuja, eikä niitä voida käyttää käytännön sovelluksissa. Tästä syystä neuroverkkoa tulee opettaa ratkaisemaan annettu tehtävätyyppi tarkemmin. Opettamisen aikana neuroverkon paino- ja vakiokertoimia muutetaan siten, että ulostulokerroksen aktivoinnit vastaavat tarkkemmin oikeita vastauksia. Neuroverkon opettaminen voidaan toteuttaa ohjatusti, ohjaamattomasti tai näiden yhdistelmällä.

Ohjatussa oppimissa (engl. supervised learning) neuroverkolle syötetään syötekerrokselle testitapauksia ja ulostulokerrokselle näiden testitapausten oikeat vastaukset. Testitapaukset ja niiden halutut ratkaisut pitää saada jostakin ennen neuroverkon opettamista, mikä voi olla kohdealueesta riippuen haastavaa. Ohjaamattomassa oppimisessa (engl. unsupervised learning) puolestaan ei tunneta esimerkkitapausten oikeita vastauksia. Siinä vaiheessa kun neuroverkko on opetettu ja se alkaa palauttamaan halutun tarkkuuksisia tuloksia, voidaan sitä hyödyntää käytännön sovelluksissa.

3 NP-vaikeat kombinatoriset optimointiongelmat

Eräs laaja matemaattinen ongelmatyyppi on erilaiset optimointiin liittyvät tehtävät. Näissä tavoitteena on minimoida tai maksimoida kohdefunktion arvo siten, että funktion parametrit ovat tiettyjen rajoitteiden mukaisia. Tällainen tehtävä ilmaistaan usein seuraavasti:

$$\begin{aligned} \underset{x}{\text{minimoi}} \quad & f(x), \\ \text{s.e} \quad & g(x) \leq 0, \\ & h(x) = 0. \end{aligned} \tag{3.1}$$

Tehtävässä (3.1) funktio $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ on tehtävän kohdefunktio, jonka optimi halutaan ratkaista, funktio $g : \mathbb{R}^n \mapsto \mathbb{R}^p$ on optimointitehtävän epäyhtälörajoite ja funktio $h : \mathbb{R}^n \mapsto \mathbb{R}^q$ on tehtävän yhtälörajoite.

Kombinatoristen optimointiongelmien tapauksessa halutaan löytää mahdollisimman optimaalinen yhdistelmä kappaleita, objekteja tai pisteitä, joilla kohdefunktion arvo olisi optimaalinen. Koska optimaalisen joukon löytämiseksi tulisi käydä läpi kaikki mahdolliset yhdistelmät, niin usein nämä tehtävät ovat NP-täydellisiä. Tämän takia monesti käytetään erilaisia heuristisia menetelmiä ratkaisun approksimoimiseen.

Seuraavaksi esitellään tämän tutkielman kannalta keskeiset optimointitehtävät ja niiden formaalit määritelmät. Ensin käsitellään kauppamatkustajan ongelma ja sen heuristiikkoja, minkä jälkeen siirrytään kuvaamaan kapsäkkiongelmaa, siihen käytettyjä heuristisia menetelmiä ja variaatioita.

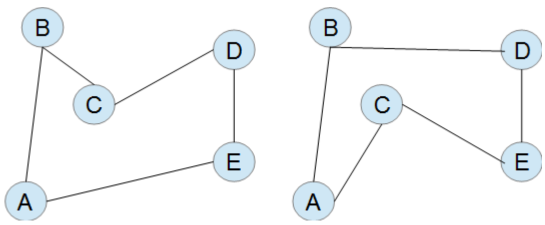
3.1 Kauppamatkustajan ongelma

Kauppamatkustajan ongelma (engl. traveling salesman problem, lyhennetään TSP) on klassinen ja hyvin määritelty esimerkki NP-täydellisestä optimointitehtävästä. Tavoitteena on löytää suljettu reitti, joka käy jokaisessa kaupungissa ja palaa takaisin aloituskaupunkiin siten, että kuljetun matkan pituus olisi mahdollisimman lyhyt. Kyseessä on NP-vaikea ongelma ja n kaupungin kaikkien reittien läpikäyminen vaatii $(n-1)!$ tarkistusta. Olkoon $G=(V,E)$ verkko, jossa V on kaikkien kaupunkien joukko ja E on kaikkien kaupunkien välisten reittien joukko. Olkoon $uv \in E$ kahden kaupungin välinen kaari ja d_{uv} tämän kaaren pituus. Tällöin kauppamatkustajan ongelma voidaan formalisoida kokonaislukuoptimointitehtävänä

$$\begin{aligned}
\text{minimoi} \quad & \sum_{\substack{u=0 \\ u\neq v}}^{n} \sum_{\substack{v=0 \\ v\neq u}}^{n} d_{uv}x_{uv}, \\
\text{s.e} \quad & \sum_{\substack{u=0 \\ u\neq v}}^{n} x_{uv} = 1, \\
& \sum_{\substack{v=0 \\ v\neq u}}^{n} x_{uv} = 1, \\
& a_u - a_j + px_{uv} \leq p-1.
\end{aligned} \tag{3.2}$$

Tämä on käsiteltynä Millerin, Tuckerin ja Zemlinin (1960) artikkelissa.

Kuviossa 2 on esimerkkikuva tyypillisestä TSP:stä, missä kaupunkijoukko V sisältää kaupungit A, B, C, D ja E. Kaupunkien välisten kaarien pituuksia ja kaikkia kaareja ei ole kuvattuna.



Kuvio 2. Esimerkkikuva tyypillisestä kauppamatkustajan ongelmasta ja kaksi mahdollista ratkaisua.

TSP:lle on myös kehitetty monenlaisia heuristisia ratkaisumenetelmiä, jotka pyrkivät löytämään optimin nopeasti. Näitä ovat muun muassa lähimmän naapurin menetelmä ja erilaiset sijoitusmenetelmät. Nilssonin (2003) raportissa on kuvattuna joitakin näistä heuristiikoista lyhyesti. Lähimmän naapurin menetelmässä (engl. Nearest Neighbor Algorithm) ideana on valita aina se kaupunki, jolla on lyhin etäisyys nykyiseen kaupunkiin verrattuna. Tätä jatketaan jostakin satunnaisesti valitusta ensimmäisestä kaupungista alkaen niin kauan, kunnes kaikki kaupungit on käyty läpi. Lopuksi palataan lähtökaupunkiin. Kyseinen ahne menetelmä on helppo ja yksinkertainen toteuttaa, mutta ottaa ratkaisuun mukaan todella pitkiä kaaria ja keskiarvoisesti tällä metodilla saadaan 1,26-kertaa suurempia arvoja optimiin nähden. Nämä tarkemmat tarkastelut ovat Cookin ym. (1998) kirjasta.

Sijoitusmenetelmien perusideana on muodostaa kaarista tilapäinen alireitti, johon ei ensimmäisellä kerralla kuulu kaikkia tehtävän kaupunkeja. Tätä alireittiä päivitetään siten, että siihen lisätään jollakin valitulla ehdolla uusia kaupunkeja. Esimerkiksi voidaan valita uusi lisättävä kaupunki siten, että käydään kaikki alireitin kaupungit läpi ja verrataan niiden etäisyyksiä alireittiin kuulumattomien kaupunkien kanssa. Valitaan se kaupunki, jolla on lyhin etäisyys johonkin näistä kaupungeista ja muodostetaan uusi alireitti siten, että sen pituus kasvaisi mahdollisimman vähän.

Suoran vastauksen antavien algoritmien lisäksi voidaan jollakin heuristiikalla saatua tulosta parantaa erilaisilla algoritmeilla. Esimerkiksi voidaan tarkastella saatua reittiä 2-opt-algoritmilla, jonka ideana on erottaa kyseinen reitti kahteen osaan ja yhdistää nämä osat siten, että uuden reitin pituus on lyhyempi. Tästä voidaan tehdä versio, jossa valitaan jokin sellainen kaarijoukko, jossa on enintään k kappaletta kaaria. Näitä kaarijoukkoja testataan vuoron perään samalla idealla kuin 2-opt-algoritmia, eli poistettujen kaarien tilalle etsitään uudet kaaret, jotka tekisivät reitistä lyhyemmän. Cookin ym. (1998) mukaan 2-opt-algoritmi tuottaa keskimäärin 1,06-kertaa suurempia arvoja optimiin verrattuna, kun taas 3-opt-algoritmi tuottaa noin 1,04-kertaa suurempia arvoja.

Kauppamatkustajan ongelmasta on olemassa useita erilaisia variaatioita. Esimerkiksi voidaan tarkastella tapausta, jossa on useampi kuin yksi kauppamatkustaja kulkemassa reittiä aloituskaupungista alkaen. Tehtävänä on minimoida näiden kauppamatkustajien yhteinen reitti siten, että kaikki kaupungit käydään läpi ja kaikki kauppamatkustajat ovat palanneet lähtökaupunkiin. Tämän tehtävän kuvaus ja mahdolliset sovellusalueet on esitelty Bektasin (2006) artikkelissa. Tämän tarkemmin ei käsitellä variaatioita, sillä suurin osa keinotekoisten neuroverkkojen käytöstä TSP:n ratkaisemiseen keskittyi alkuperäiseen edellä mainittuun tapaukseen.

3.2 Kapsäkkiongelma

Kapsäkkiongelma (engl. knapsack problem) on toinen tyypillinen esimerkki kombinatorisessa optimoinnissa. Tehtävänä on optimoida esineiden yhteinen arvo siten, että niiden yhteispaino ei ylity tietyn painorajan yli. Kapsäkkiongelma on NP-vaikea (Korte, Vygen 2012). Ongelma voidaan kuvata kokonaislukuoptimointitehtävänä

$$\begin{aligned} &\text{maksimoi} && \sum_{i=1}^{n} v_i x_i, \\ &\text{s.e} && \sum_{i=1}^{n} w_i x_i \leq c, \end{aligned} \tag{3.3}$$

missä $v_i \in \mathbb{R}$ on esineen $i \in \{1, ..., n\}$ arvo(englannista *value*, monesti käytetään myös merkkiä p_i sanasta *profit*), $w_i \in \mathbb{R}$ sen esineen paino ja $x_i \in 0, 1$ kuvaa esineen mukaanottamista siten, että $x_i = 0$ niin esinettä ei oteta mukaan, $x_i = 1$ niin otetaan. Tästä syystä tätä tehtävätyyppiä kutsutaan usein 0-1 -kapsäkkiongelmaksi.

Tästä voidaan vielä tehdä niin sanottu LP-relaksaatio poistamalla x_i:n kokonaislukurajoite muotoon $0 \leq x_i \leq 1$, $x_i \in \mathbb{R}$, jolloin tehtävän muoto muuttuu jatkuvaksi ja tehtävän ratkaisussa voidaan yrittää hyödyntää lineaaristen optimointitehtävien ratkaisualgoritmeja, esimerkiksi Simplex-algoritmia. Tätä LP-relaksaatiosta saatua tehtävää kutsutaan jatkuvaksi kapsäkkiongelmaksi (Martello, Pisinger, Toth 2000). Lisäksi tehtävän esineet voidaan järjestää niiden arvojen ja painojen suhteen mukaan siten, että

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \cdots \geq \frac{v_n}{w_n}. \tag{3.4}$$

Tälle kapsäkkiongelmalle on olemassa useita erilaisia ratkaisumenetelmiä, joista osa perustuu kyseisen ongelman muuttamiseksi tällaiseen muotoon. Esimerkiksi Martello ja Toth (2000) ovat kuvailleet ahneen menetelmän, jossa otetaan mukaan esineitä (3.4) mukaisessa järjestyksessä niin kauan, kunnes seuraava esine ylittäisi kapsäkin painorajan c. Tätä kyseistä esinettä kutsutaan kriittiseksi esineeksi. Samalla tämänmuotoisesta tehtävästä voidaan arvioida esineiden yhteisarvon ylärajaksi

$$\sum_{j=1}^{s-1} v_j + c - \sum_{j=1}^{s-1} w_j v_s w_s, \tag{3.5}$$

missä s on kriittisen esineen indeksi.

0-1 -kapsäkkiongelmasta voidaan tehdä laajempi tehtävätyyppi muodostamalla kvadrattiinen kapsäkkiongelma (engl. quadratic knapsack problem). Tämän optimointitehtävän kuvaus on

$$\begin{aligned} &\text{maksimoi} && \sum_{i=1}^{n}\sum_{j=1}^{n} v_{ij} x_i x_j \\ &\text{s.e} && \sum_{j=1}^{n} w_j x_j \leq c, \end{aligned} \tag{3.6}$$

missä v_{ij} on esineiden i ja j yhteisarvo, ja se kuuluu $n \times n$ -kokoiseen ei-negatiivisten kokonaislukujen matriisiin. Muuttujat x_j ja w_j ovat vastaavia kuin 0-1 -kapsäkkiongelmassa. Tämä tehtävä palautuu takaisin 0-1 -KP:ksi tapauksessa, jossa $v_{ij} = 0$ kaikilla $i \neq j$. Kaavan 3.6 esittely ja kyseisen ongelman tarkka ratkaiseminen on esitetty Capraran, Pisingerin ja Tothin (1999) artikkelissa.

4 Optimointiongelmien ratkaiseminen keinotekoisten neuroverkkojen avulla

Tässä luvussa tarkastellaan aiemmissa luvuissa käsiteltyjä neuroverkkoja esitettyjen tehtävien ratkaisemiseen ja selvitetään kuinka hyvin ANN:t soveltuvat tähän sovelluskohteeseen. Tarkastelemme tämän aiheen tutkimusta alusta alkaen ja käsittelemme nykyaikaisia tutkimustuloksia. Ensin käsitellään neuroverkkojen käyttöä kauppamatkustajan ongelmaan ja sen jälkeen kapsäkkiongelman tapauksessa. TSP:tä käsittelevät artikkelit testasivat keinotekoisia neuroverkkojaan TSPLIB-kirjaston ongelmiin.

Tutkimuksia neuroverkkojen käyttämisestä NP-vaikeiden optimointitehtävien ratkaisemisesta on jo vuonna 1985 tehdyssä Hopfieldin ja Tankin kirjoittamassa artikkelissa. Kyseisessä tutkimuksessa selvitettiin analogisten neuroverkkojen käyttöä vaihtoehtoisena menetelmänä ratkaista vaikeita optimointitehtäviä. He demonstroivat ratkaisumenetelmäänsä kauppamatkustajan ongelmalla. Kyseisessä menetelmässä TSP esitettiin kaksiulotteisena taulukkona, jossa jokainen rivi kuvaa tiettyä kaupunkia ja jokainen sarake vastaa kaupungin sijaintia reitissä. Solut vastaavasti kuvaavat tietyn kaupungin valintaa sarakkeen kuvaamassa sijainnissa. Reitin valinta on jonkin solun liukuluku nollan ja ykkösen väliltä.

Hopfieldin ja Tankin kehittämän neuroverkon rakenteessa on myös olennaisena osana energiafunktio, jonka ideana on saada neuronin arvot suppenemaan kohti jotain arvoa joka kerta, kun neuronien arvoja päivitetään. Joka kerta kun neuronien tilaa päivitetään, energiafunktion arvo joko pysyy samana tai vähenee. Valitettavasti Hopfieldin ja Tankin metodissa käytetty energiafunktio saattaa tuottaa epäkelvollisia tuloksia lähtöarvojen valinnasta riippuen. Lisäksi Geen ja Pragerin (1995) mukaan satunnaisesti luotujen kauppamatkustajan ongelmien ratkaisemisessa Hopfieldin ja Tankin neuroverkkoon perustuvilla menetelmillä ei välttämättä saatu minkäänlaista ratkaisua, tai ne olivat noin 40 % hitaampia 3-opt-algoritmiin verrattuna. Hopfieldin ja Tankin artikkelissa on kuitenkin ensimmäisenä kuvattu keinotekoinen neuroverkko, jolla voidaan saada TSP:lle ratkaisu.

Vuosien 1985 ja 1999 välillä on ollut jonkun verran lisää tutkimusta aiheesta. Smith (1999) on tehnyt tästä koosteen, jossa kerrataan ja esitetään erityyppisten optimointitehtävien ratkaisemista keinotekoisten neuroverkkojen avulla. Artikkeliin on kerättynä monia kombinatorisia optimointitehtäviä ja niihin käytettyjä neuroverkkopainoitteisia ratkaisuja. Kyseisessä artikkelissa kuitenkin todetaan, että monet sen aikaisista tutkimuksista eivät tehneet vertailuja muiden optimointimenetelmien välillä, mutta myöhemmät tutkimukset ovat vertailleet metodejaan aikaisempiin menetelmiin nähden.

Saman artikkelin (1999) mukaan neuroverkkojen käyttöä kombinatoristen optimointiongelmien ratkaisemisessa on selvästi testattu useammin kauppamatkustajan ongelmaan muihin optimointitehtäviin verrattuna. Tämä johtunee Smithin mukaan siitä, että TSP:lle on kehittynyt eräänlainen "mittapuun"maine keinotekoisiin neuroverkkoihin perustuvien ratkaisumenetelmien testaamisessa. Smith (1996) on myös kirjoittanut artikkelin, jossa hän väittää TSP:n olevan väärinkäytetty mittapuu neuroverkkojen tehokkuuden tarkistamisessa yleisenä optimointitekniikkana.

Masutti ja Castro (2009) käyttivät itsestään järjestäytyviä neuroverkkoja TSP:n ratkaisemiseen. Heidän artikkelissaan on kuvattu muokatun RABNET-TSP:n käyttöä samalla vertaillen sen antamia tulosten tehokkuutta kolmen muun algoritmin tehokkuuteen. Alkuperäinen RABNET-TSP (*real-valued antibody network for the TSP*) on kuvailtu Pastin ja Castron (2006) aikaisemmassa artikkelissa. Kyseinen neuroverkko on eteenpäinkytketty, mutta ei sisällä ollenkaan piilokerrosta. Ideana on muodostaa RABNET siten, että sen neuronit kuvautuvat suoraan ratkaistavan TSP:n päälle, jolloin neuronien jono vastaisi mahdollista ratkaisua. RABNET myös kasvaa ohjaamattoman oppimisen aikana kehämäisesti, jolloin sen tuottamaan reittiin ei luonnostaan tule yhtään ristikkäisiä kaaria.

Alle viidensadan kaupungin ongelmissa artikkelin algoritmilla saadut tulokset erosivat parhaasta ratkaisusta alle 1 % ja suuremmalla kaupunkilukumäärällä oli enintään 14 % eroa parhaista ratkaisuista. Samalla yli tuhannen kaupungin ongelmissa artikkelin RABNET-TSP algoritmi vaati vähemmän laskentaa, mutta sen antamat ratkaisut eivät olleet nopeampia ratkaisemaan artikkelissa testattuja ongelmia. Mahdollisena jatkoideana Masutti ja Castro ehdottavat RABNET-TSP:n ja reitinparannus -algoritmien, kuten k-opt-algoritmin yhdistämistä.

Ohlsson, Peterson ja Söderberg (1993) kehittivät ANN:iä hyödyntävän menetelmä ratkaisemaan epäyhtälörajoitteisia optimointitehtäviä ja demonstroivat tätä kapsäkkiongelmalla. Kyseisessä artikkelissa oleva kapsäkkiongelma muodostui arvo- ja painokertoimista, jotka olivat väliltä $[0, 1]$ ja kapsäkin painoraja oli $\frac{N}{2}$, missä N vastaa esineiden lukumäärää. Artikkelissa käytetty neuroverkko oli rakenteeltaan takaisinkytketty (engl. feedback), eli ulostulokerroksen neuronien tulokset kytketään takaisin syötekerroksen neuroneille. Sopivilla parametreilla tämän neuroverkon tulokset saadaan suppenemaan lokaalia optimia kohden. Muihin ratkaisumenetelmiin verrattuna artikkelin neuroverkko laski ratkaisut nopeammin silloin, kun kapsäkkiongelman arvokertoimet olivat homogeeniset.

Myös yleiselle kvadraattiselle kapsäkkiongelmalle(engl. generalized quadratic knapsack problem, lyhennetään GQKP) on kehitetty keinotekoisiin neuroverkkoihin perustuva menetelmä. Talavánin ja Yáñezin (2006) artikkelissa on muodostettu jatkuva Hopfieldin verkko, jonka neuroneille voidaan kuvata kaikki ne optimointitehtävät, jotka on esitettävissä GQKP:nä. Tämän neuroverkon tehokkuutta testattiin kauppamatkustajan ongelmaan, joka on erikoistapaus GQKP:stä. Suurin testattu TSP oli kooltaan 1002 kaupungin kokoinen, mutta valitettavasti tuloksia ei vertailtu muiden heuristiikkojen kanssa. Kaikki neuroverkon antamat ratkaisut kuitenkin tuottivat kelvollisen reitin.

5 Yhteenveto

Tässä kandidaatintutkielmassa on käyty läpi neuroverkkojen perusteista teoriaa, kombinatorisia optimointiongelmia ja niiden ratkaisemista, sekä neuroverkkojen käyttöä TSP:n ja kapsäkkiongelman ratkaisemiseksi. Aiheesta on tehty runsaasti tutkimusta Hopfieldin ja Tankin artikkelin julkaisun jälkeen, mutta kehitys on ollut välillä hidasta. Esimerkiksi pelkästään sopivan energiafunktion löytäminen on ollut haastavaa, jolloin keinotekoisen neuroverkon saadut ratkaisut eivät välttämättä ole kelvollisia ratkaisuja. Lisäksi satunnaiset tehtävät saattavat aiheuttaa ongelmia ratkaisun löytämisessä, kuten Geen ja Pragerin artikkelissa on todettu. Myös ANN:n tulokset pitää saada suppenemaan, tai muutoin sen tuottamat ratkaisut eivät lähesty mitään optimia. Mutta mikäli neuroverkon tulokset ovat kelvollisia ja ne suppenevat kohti optimia, niin sen tuottamat ratkaisut ovat olleet usein muihin metodeihin verrattuna hyviä.

Parempien keinotekoisten neuroverkkojen lisäksi olisi mielenkiintoista nähdä, että kuinka paljon esimerkiksi reitin parantamiseen perustuvat algoritmit muuttavat tulosta. Masutti ja Castro mainitsivat tästä artikkelissaan, mutta monet muut tutkijat ovat usein käsitelleet pelkästään ANN:ien käyttöä, eivätkä ole maininneet muiden heuristiikkojen yhdistämistä neuroverkkojen kanssa. Joka tapauksessa tämänhetkiset tulokset vaikuttavat erittäin lupaavilta jo pelkästään TSP:n ja kapsäkkiongelman kannalta.

Kirjallisuutta

Bektas, T. 2006. *The multiple traveling salesman problem: an overview of formulations and solution procedures*. Omega, Volume 34, Issue 3, s. 209-219.

Caprara, A. & Pisinger, D. & Toth, P. 1999. *Exact solution of the quadratic knapsack problem*. INFORMS Journal on Computing, Volume 11, s. 125-137.

Cook. W. J. & Cunningham, W. H & Pulleybank, W. R. & Schrijver, A. 1998. *Combinatorial optimization.* New York: Wiley.

DeepArt Team. 2018. *DeepArt.io - become a digital artist*. Saatavilla WWW-muodossa <URL: `https://deepart.io/`>. Viitattu 11.3.2018.

Gee, A. H. & Prager, R. W. 1995. *Limitations of neural networks for solving traveling salesman problems*. IEEE Transactions on Neural Networks Volume 6, Issue 1, s. 280-282.

Hopfield J. J. & Tank D. W. 1985. *"Neural"Computation of Decision in Optimization Problems*. Biological Cybernetics(1985) Volume 52, s. 141-152.

Korte, B. & Vygen, J. 2012. *Combinatorial Optimization: Theory and Algorithms*.

Martello, S. & Pisinger, D. & Toth, P. 2000. *New trends in exact algorithms for the 0-1 knapsack problem*. European Journal of Operational Research(2000), Volume 123, Issue 2, s. 325-332.

Masutti, T. A. S. & Castro, L. N. D. 2009. *A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem*. Information Sciences, Volume 179, Issue 10, s. 1454-1468.

McCulloch, W. S. & Pitts, W. 1943. *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematical Biophysics(1943) 5: s. 115-133.

Miller, C. E. & Tucker, A. W. & Zemlin, R. A. 1960. *Integer Programming Formulation of Traveling Salesman Problems*.

Nilsson, C. 2003. *Heuristics for the traveling salesman problem*. Linköping University.

Ohlsson, M. & Peterson, C. & Söderberg, B. 1993. *Neural networks for optimization problems with inequality constraints: the knapsack problem*. Neural Computation, Volume 5, Issue 2, s. 331-339.

Pasti, R. & Castro, L. N. D. 2006. *A Neuro-Immune Network for Solving the Trave-*

ling Salesman Problem. Proceedings of International Joint Conference on Neural Networks, s. 3760-3766.

Talaván, P. & Yáñez, J. 2006. *The generalized quadratic knapsack problem. A neuronal network approach*. Neural Networks, Volume 19, Issue 4, s. 416-428.

Rosenblatt F. 1958. *The perceptron: a probabilistic model for information storage and organization in the brain*. Psychological review, Volume 65, Issue 6, s. 386-408.

Silver, D. ym. 2017. *Mastering the game of Go without human knowledge*.

Smith, K. A. 1996. *An Argument for Abandoning the Traveling Salesman Problem as a Neural-Network Benchmark*. IEEE Transactions of Neural Networks 7, s. 1542-1544.

Smith, K. A. 1999. *Neural Networks for Combinatorial Optimization: A Review of More Than a Decade of Research*. INFORMS Journal on Computing, Volume 11, Issue 1, s. 15-34.

Villarubia, G. ym. 2018. *Artificial neural networks used in optimization problems*. Neurocomputing, Volume 272, s. 10-16.