



Katsaus Algoritmiseen säveltämiseen (David Copen johdolla)

Sami Haukka

OPINNÄYTETYÖ
Toukokuu 2020

Muusikko (AMK)
Säveltäjä

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Musiikin tutkinto-ohjelma
Muusikko (AMK): säveltäjä

HAUKKA, SAMI

Katsaus Algoritmiseen säveltämiseen (David Copen johdolla)

Opinnäytetyö 45 sivua, joista liitteitä 5 sivua
Toukokuu 2020

Tietokoneen avustuksella säveltäminen on kiinnostava aihe, ja algoritmisen säveltäminen ei terminä välttämättä ole tuttu monelle sävellystä opiskelevalle. Näiden syiden vuoksi opinnäytetyön lähtökohdaksi otettiin algoritmiseen säveltämiseen tutustuminen. Pienen, internetin hakukoneilla tehdyn, tiedonhaun jälkeen löytyi lupaavan tuntuisia kirjoja ja artikkeleita. Nimi David Cope esiintyi useissa löydetyistä lähteistä. Tämä löytö viittasi siihen, että David Cope on merkittävä henkilö algoritmisen säveltämisen saralla.

Selvisi, että David Cope on tehnyt merkittävää 'pioneerityötä' liittyen algoritmiseen säveltämiseen. David Cope on klassisen musiikin säveltäjä, mutta hän on tehnyt algoritmiseen säveltämiseen liittyvistä tietokoneohjelmistaan sellaisia, että niiden käyttämien kappaleiden tyyliä ei tarvitse olla välttämättä klassista musiikkia. Nämä ohjelmat tarvitsevat vain samaa musiikin tyyliä sisältävän tietokannan. Etsittäessä tietoa algoritmisesta säveltämisestä saatiin muodostettua parempi käsitys siitä, mitä algoritmisen säveltäminen on ja mitä algoritmisessa sävellysprosessissa tapahtuu (tietokoneen ja/tai ihmisen toimesta). Yleensä tarvitaan 1) tietokanta samantyyllisiä kappaleita, 2) ohjelma, joka analysoi tätä tietokantaa ja muodostaa omat (musiikilliset)säännöt miten tyyliä kappaleita tulisi säveltää ja 3) ihminen, joka ohjaa sävellysprosessia tarvittaessa 'oikeaan' suuntaan. Mainittakoon vielä, että musiikin tyyliä ei ole periaatteessa merkitystä algoritmisessa säveltämisessä, jos säveltämisen apuna käytettävä ohjelmisto muodostaa taustalla olevasta tietokannasta omat sääntönsä.

Opinnäytetyö syntyi tekijän omasta mielenkiinnosta algoritmista säveltämisestä kohtaan. Sävellystä opiskeleville olisi hyödyksi tietää, mitä algoritmisen säveltäminen on. Esimerkiksi tekoälytekniikoita tullaan hyödyntämään enemmän musiikkiteollisuudessa kuin aikaisemmin – kuten on jo käynyt. Sävellysohjelmoijoiden olisi siis hyvä tietää näistä asioista edes jonkin verran. Yksi hyvä tapa olisi tarjota suomalaisissa yliopistoissa ja ammattikorkeakouluissa kursseja liittyen algoritmiseen säveltämiseen ja tekoälytekniikoihin. Toivottavasti tästä opinnäytetyöstä on hyötyä vasta algoritmiseen säveltämiseen tutustuvalla ja jo aihepiiriin hieman enemmän perehtyneelle lukijalle.

Asiasanat: algoritmisen säveltäminen, tietokoneavusteinen säveltäminen, algoritmi, sävellys, David Cope

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Program in Culture and Arts, Music
Option of Composing

HAUKKA, SAMI:

A brief look at algorithmic composing (led by David Cope)

Bachelor's thesis 45 pages, appendices 5 pages

May 2020

Composing with the help of a computer is an interesting topic, and algorithmic composing as a term is probably unknown to many students learning composition. For these reasons algorithmic composing was selected to be the subject of this thesis. After some research, using the internet search sites, some promising articles and books were found. The name, David Cope, featured prominently in the researched literature, which caught the author's attention and suggested that David Cope probably is an important person in the field of algorithmic composing.

It was found out that David Cope had done a lot of pioneer work in the field of algorithmic composing. David Cope is mainly a classical music composer, but he has developed his algorithmic composing related computer programs in a way that does not demand pieces to be in the classical music style. His algorithmic composing programs only need musical pieces of the same style in the database they are using. When the author searched information about algorithmic composing, better understanding about the procedures used was achieved. Typical elements include 1) a database that has pieces in the same musical style, 2) a program that analyses this data and creates musical rules about how to compose in the style used in the database and 3) a human that guides the composing process to the right direction (if needed). It does not matter what the musical style is, if the computer program itself generates the rules specific to that style as a result of its database analysis.

The work originated from the author's own interest in algorithmic composing. In the author's opinion, it would be beneficial for composition students to learn about algorithmic composing. For example, artificial intelligence will affect the music industry – and already has. Composers should be familiar with it. One possibility would be that universities in Finland would offer courses related to algorithmic composing and artificial intelligence techniques to composition students. It is hoped that the reader can benefit from this thesis whether or not the reader is just starting to learn about algorithmic composing or already has some background knowledge.

Key words: algorithmic composing, algorithmic composition, algorithmic composer, algorithm, composition, David Cope

SISÄLLYS

JOHDANTO	9
1 ALGORITMISEN SÄVELTÄMISEN MÄÄRITELMÄ JA MUSIIKIN HISTORIAA	10
1.1 Algoritmisen säveltämisen määritelmä	10
1.2 Algoritminen säveltäminen musiikin historiassa	10
1.2.1 David Cope ja algoritminen säveltäminen	11
1.3 Algoritmisen säveltämisen määritelmä tulevaisuudessa?	12
2 TEKOÄLYN VALLANKUMOUS	13
2.1 Tekoäly on todellisuutta jo nyt	13
2.1.1 Syväoppiminen	14
2.1.2 Mitä syväoppiminen oikeastaan on?	15
2.1.3 Syväoppimiseen liittyviä ongelmia	16
2.2 Tekoäly käytössä tänä päivänä	17
2.3 Tekoäly ja sen suhde algoritmiseen säveltämiseen?	18
3 MITÄ ALGORITMISESSA SÄVELTÄMISESSÄ TAPAHTUU	19
3.1 "Kerää-analysoi-luo" -lähestymistapa	19
3.1.1 Datan kerääminen	19
3.1.2 Datan analysointi	20
3.1.3 Sävellyksen tekeminen	20
3.2 Esimerkkitapaus popmusiikista	20
3.2.1 Popmusiikkia tietokoneen säveltämänä	20
3.3 Tietokone säveltää barokkimusiikkia YouTube-video	23
3.4 Käytettyjä menetelmiä ja ohjelmistoja?	23
4 ALGORITMISEN SÄVELTÄMISEN MENETELMIÄ JA OHJELMISTOJA	25
4.1 Lähestymistapoja ja vaikuttamiskeinoja	25
4.1.1 Stokastisuus ja todennäköisyydet	26
4.1.2 Markovin ketjut	27
4.1.3 'State machines' eli tilakoneet	28
4.1.4 Muita menetelmiä	28
4.2 Joitakin ohjelmistoja	28
4.2.1 AIVA-ohjelmisto	28
4.2.2 Flow Machines-ohjelmisto	29
4.3 YouTube opetusvideoita	29
4.4 Historian havinaa - 2000-luvun alku	29
5 DAVID COPEN OHJELMISTOJA	31

5.1.1 David Copen Emi-ohjelma	31
5.1.2 David Copen Alice-ohjelma	33
5.2 David Copen aikaansaannoksia	35
5.3 Muiden henkilöiden aikaansaannoksia	35
5.4 Algoritmisen säveltämisen hyödyt ja haitat?	36
6 ALGORITMISEN SÄVELTÄMISEN HYÖTYJÄ	37
6.1 Vertailu eri säveltäjien teosten kesken	37
6.2 Musiikkityyliin 'sisään' pääseminen helpottuu	37
6.3 Haarautumakohdassa mahdollisen jatkon valitseminen helpottuu	37
6.4 Tietokone saa luotua lähtödataansa paremman teoksen	38
7 ALGORITMISEN SÄVELTÄMISEN HAITTOJA	39
7.1 Tietokoneen käyttö voi vähentää säveltäjän omaa ajatustyötä	39
7.2 Tietokone voi luoda ison määrän kappaleita sekunneissa	39
7.3 "Tietokone vastaan ihminen"-vastakkain asettelu	40
8 VASTAUKSIA JOHDANTO-LUVUN KYSYMYKSIIN	41
9 POHDINTA	45
LÄHTEET	46
LIITTEET	47
Liite 1. David Copen luento "Computer synthesis of musical creativity"	
47	
Liite 2. David Copen tekemiä musiikkikappaleita ja kirjoja	48
Liite 3. Algoritmisen säveltämisen seurauksena syntyneitä kappaleita.	
50	
Liite 4. YouTube opetusvideoita	51

LYHENTEET JA TERMIT

algoritminen säveltäminen	Jonkin algoritmin hyödyntämistä sävellysprosessin aikana. Voi olla joko tietokoneen avulla tehty tai vaikka kynällä ja paperilla hahmoteltu kappaleen rakenne. Olennaista askelittain tapahtuva ohjeen seuraaminen, kunnes päästään lopputulokseen.
Alice	“ALgorithmically Integrated Composing Environment”. David Copen tekemä algoritmiseen säveltämiseen liittyvä tietokoneohjelma. Emi-ohjelman ’jälkeläinen’.
Bayesilainen (neuro)verkko	Bayesilainen verkko on verkkotodennäköisyysmalli, joka esittää joukon satunnaisuuttuja ja niiden välisiä ehdollisia riippuvuuksia suunnatun syklittömän verkon avulla.
barokki	Termi, jolla tarkoitetaan 1600-luvulla ja 1700-luvun alkupuolella sävellettyä länsimaista taidemusiikkia. Ajanjaksona barokki sijoittuu noin vuosien 1600-1750 väliin.
DAW	Digital Audio Workstation. Tarkoittaa tietokoneohjelmaa, kuten Cubase tai Apple Logic Pro, joka on tarkoitettu auttamaan musiikin tekijää. Muun muassa äänen editointi ja miksaus onnistuu näillä ohjelmilla.
deterministinen	Ei-satunnainen vaan jokainen askel/siirtymä on ennalta määritelty.
diskreetti	Äärellinen. Esimerkiksi luonnolliset luvut ovat diskreettejä.
Emi	“Experiments in Musical Intelligence”. David Copen tekemä algoritmiseen säveltämiseen liittyvä tietokoneohjelma. Alice-ohjelman ’äiti’.

formaali menetelmä	Formaali eli muodollinen. Matematiikassa formaali on saanut erityismerkityksen 'mekaanisesti esimerkiksi tietokoneella tarkistettavissa oleva'. Kaiken kattavaa määritelmää on vaikea löytää, mutta yhteistä kaikille formaaleille menetelmille on matemaattisen täsmällisyyden korostaminen.
fuuga	Latinaksi fuga eli suomeksi pako tai paeta. Kontrapunktia käyttäen sävelletty polyfoninen musiikkikappale tai kappaleen osa, joka koostuu useasta itsenäisestä äänestä.
isorytmiikka	Keskiajan musiikissa syntynyt sävellystekniikka, jossa melodiaa esittävä ääni koostuu samanlaisina toistuvista rytmillisistä kuvioista.
kaanon	Suomeksi laki. Kaanon on kontrapunktinen sävellys, jossa eri äänet esittävät samaa melodiaa tai sen kontrapunktista johdosta siten, että ne aloittavat sen eri aikaan ja seuraavat sitten toisiaan tietyllä viiveellä.
kontrapunkti	Latinaksi "punctum contra punctum". Suomeksi "nuotti nuottia vastaan". Kahden tai useamman melodian hallittua yhdistämistä yhtäaikaiseksi.
MIDI	Musical Instrument Digital Interface. Musiikkisoittimien digitaalinen liitäntä tai rajapinta. Tiedonsiirtojärjestelmä, joka on suunniteltu välittämään viestejä sähköisten musiikkilaitteiden välillä.
moodi	Pysyvä musiikin rakenne kuten asteikko, melodia tai pysyvä rytmikuvio. Länsimaisessa musiikissa tarkoitetaan usein keskiajalla kehitettyä asteikkojärjestelmää eli niin sanottuja kirkkosävellajeja.

motetti	Erilaisista kuoroteoksista käytetty nimitys. Motetit olivat merkittäviä polyfonisen musiikin muotoja 1200-luvulta noin 1700-luvun puoliväliin saakka.
renessanssi	Keskiajan jälkeen Euroopassa palattiin antiikin ihanteisiin. Noin 1300-1600.
sarjallisuus	Toiselta nimeltään serialismi. Modernin musiikin menetelmä teosten säveltämiseen ja analysointiin, jossa eri musiikin osat, kuten äänenkorkeus ja dynamiikka, järjestetään sarjaksi.
stokastinen	Satunnainen. Ajassa sattumanvaraisesti etenevä (matemaattinen) prosessi.
syväoppiminen	Englanniksi ”deep learning”. Tekoälyn oppimismenetelmä, jossa hyödynnetään usean kerroksen neuroverkkoja.
tekoäly	Keinoäly. Tietokone tai tietokoneohjelma, joka kykenee tekemään älykkäinä pidettyjä toimintoja.
12-säveltekniikka	Sävellystekniikka, jossa sävelasteikon 12 säveltä on määritelty tasa-arvoisiksi.

JOHDANTO

Algoritminen säveltäminen (tai terminä samaa tarkoittava tietokoneavusteinen säveltäminen) kiinnostaa minua, koska olen pohtinut muun muassa seuraavia kysymyksiä musiikin opintojeni aikana:

1. Miten tietokone osaa säveltää? Osaako se säveltää?
2. Algoritminen/Tietokoneavusteinen säveltäminen... Mitä se on? Ja mitä se ei ole?
3. Pystyykö ihminen erottamaan (helposti) tietokoneen tekemän sävellyksen?
4. Jos tietokone osaa säveltää yhtä hienoa musiikkia kuin parhaimmat ihmissäveltäjät (kuten Mozart, Beethoven), tarvitaanko ihmistä/säveltäjää lopulta enää sävellysprosessissa ollenkaan?
5. Kuinka nopeasti tietokone säveltää kappaleen valmiiksi? Meneekö siihen sekunteja, tunteja, päiviä, viikkoja, kuukausia vai jopa vuosia?
6. Kuinka paljon ihminen voi vaikuttaa tietokoneen sävellysprosessiin?
7. Mitä ohjelmia on jo olemassa algoritmiseen säveltämiseen liittyen?
8. Onko tekoäly (englanniksi 'artificial intelligence' eli AI) korvaamassa/syrjäyttämässä myös säveltäjien ammattikunnan jossain vaiheessa? Onhan esimerkiksi tavallisia rutiininomaisia töitä, kuten pellon auraus, jo teollisen vallankumouksen alusta siirretty ihmiseltä koneen avustuksella tehtäväksi.
9. Onko algoritmisesta säveltämisestä hyötyä säveltäjälle? Tulisiko tulevaisuuden säveltäjien tutustua tietokoneavusteiseen säveltämiseen enemmän, jotta työpaikka löytyisi AI-murroksen keskellä?

Toivon voivani vastata näihin kysymyksiin tässä opinnäytetyössä niin, että lukija saisi paremman käsityksen tietokoneen mahdollisuuksista ja rajoituksista algoritmiseen säveltämiseen liittyen.

1 ALGORITMISEN SÄVELTÄMISEN MÄÄRITELMÄ JA MUSIIKIN HISTORIAA

Tässä luvussa tutustutaan David Copen määritelmään siitä, mitä algoritmisen säveltäminen on ja mitä sillä tarkoitetaan. Samalla mainitaan myös joitakin (musiikin) historian merkittäviä henkilöitä ja tapahtumia algoritmiseen säveltämiseen liittyen.

1.1 Algoritmisen säveltämisen määritelmä

David Cope kertoo algoritmisen säveltämisen koostuvan 1) algoritmeista ja 2) säveltämisestä. Hän toteaa algoritmin olevan joukko askelia. Näitä askeleita otetaan yksi kerrallaan, kunnes saavutetaan haluttu lopputulos/maali. Esimerkkeinä algoritmeista hän mainitsee muun muassa DNA:n ja lautapeliä pelaamisen. Tietokoneella suoritettavat algoritmit noudattavat samaa logiikkaa - eli tehdään tehtävä askel kerrallaan, kunnes se on suoritettu. Algoritmisen säveltäminen on Copen mukaan yksinkertaisesti sitä, että tietokoneella sävelletään uusi kappale askel kerrallaan, kunnes sävellys on valmis. (Muscutt 2007)

Cope jatkaa, että algoritmi voi olla löysästi määritelty, kuten esimerkiksi sonaattimuoto, tai tarkasti määritelty, kuten barokkiajan kenraalibasso (Muscutt 2007). Tällä hän tarkoittaa luultavasti sitä, että sonaattimuodossa rakenne on aika tarkasti määrätty, mutta sisältö voi vaihdella – esimerkiksi melodia. Barokkiajan kenraalibassossa taas on kuitenkin aika pieni liikkumavara sointujen käänöksien/aseman suhteen. Tarkasti määritellyssä algoritmessa vaiheita seurataan tarkasti ohjeiden (eli käskyjen) mukaan. Löysästi määritellyssä algoritmessa on varaa vaihtoehtoisille toimintatavoille.

1.2 Algoritmisen säveltäminen musiikin historiassa

Copen mielestä historian tärkeimpiä säveltäjiä on 1300-luvulla elänyt säveltäjä Guillaume de Machaut. Machaut nimittäin käytti isorytmiikkaa säveltäessään messuja, motetteja ja muuta uskonnollista musiikkia. Myös Johann Sebastian Bach on Copen mukaan oleellinen historiallinen henkilö tekemillään fuugilla, kaanoneilla ja vastaavilla algoritmisilla sävellyksillään. (Muscutt 2007)

Esimerkkinä algoritmista David Cope mainitsee muun muassa Johan Joseph Fuxin kokoamat kontrapunktisäännöt (Cope 2000, 6.). Cope kertoo myös mielihiteenään, että Mozart, säveltäessään melodioita, käytti sääntöjä/algoritmeja pohjana. Copen mielestä Mozartin suuruus ja nerous tulee esille niissä tilanteissa, jolloin Mozart vältti tavanomaisia ratkaisuja. (Cope 2000, 10.)

Hanna Järveläinen (2000) kertoo, että länsimaisen musiikin historian aikana on käytetty formaaleja menetelmiä – enemmän tai vähemmän. Formalismi on taustalla vaikuttamassa monessa asiassa musiikissa. Esimerkiksi 1000-luvulla Guido Arezzolainen käytti menetelmää, jossa jokaiselle tavulle määriteltiin oma sävelkorkeutensa. 1400-luvulla käytettiin isorytmisiä motetteja. Guillaume Dufay käytti kultaista leikkausta järjestääkseen sävellyksiensä muodon. Renessanssin ja barokin aikana määriteltiin tiukat säännöt kontrapunktia varten. 1900-luvun alussa Arnold Schönberg alkoi käyttää 12-säveltekniikkaa ja tästä kehittyi sarjallisuus Anton Webernin ja hänen seuraajiensa toimesta. Sarjallisessa tekniikassa musiikin parametrit, kuten kesto, äänenkorkeus ja voimakkuus, asetetaan (tiukkojen) sääntöjen alaisuuteen. 1950-luvun aikana Iannis Xenakis alkoi kokeilla stokastisia prosesseja musiikin saralla ja teki niiden avulla sävellyksiä ilman tietokoneen apua. Ensimmäinen kokonaan tietokoneen säveltämä kappaletta, ”The Iliac Suite for String Quartet”, syntyi 1956. (Järveläinen 2000, 1.)

1.2.1 David Cope ja algoritminen säveltäminen

David Cope kertoo kirjansa ”The Algorithmic Composer” esipuheessa unelmaakaan. Hänen haaveensa oli, että hän saisi napinpainalluksella luotua musiikkia silmänräpäyksessä. Tämä musiikki olisi hänen tyylilleen uskollista ja se sopisi juuri hänen työstämäänsä teokseen. Cope jatkaa, että ”The Algorithmic Composer” -kirja on kolmas osa hänen kirjoittamastaan trilogiasta: kirja kertoo yksityiskohtaisesti, miten tietokonetta käytetään algoritmisessa säveltämisessä apuna.

Cope mainitsee esipuheessaan lisäksi Alicen (ALgorithmically Integrated Composing Environment). Alicen avulla säveltäjä voi muun muassa 1) pyytää jatkoa tekemäänsä teoksen osaan, 2) saada apua, jos on mennyt ’lukkoon’ (eli ei keksi mitään säveltää) ja 3) tutkia mitä ehdotuksia ohjelma tarjoaa sille annetun musiikillisen aiheen (esimerkiksi teeman) jatkoksi. Alicen idea on olla säveltäjän apuna

– ei niinkään säveltää 'itse'. Alice pohjaa vahvasti käyttämäänsä tietokantaan, ja jos säveltäjä täyttää tietokannan omilla sävellyksillään, pitäisi säveltäjän saada tietokoneelta uutta säveltäjän tyyllille uskollista musiikillista materiaalia. (Cope 2000)

Cope on sitä mieltä, että jokainen säveltäjä voidaan nähdä algoritmisenä säveltäjänä siinä mielessä, että jokainen säveltäjä noudattaa jonkinlaista algoritmia säveltäessään – oli se sitten tiedostettu tai tiedostamaton ratkaisu. Näin ollen Cope näkee, että säveltäjät voisivat hyötyä nyt ja tulevaisuudessa tietokoneavusteisesta säveltämisestä. (Cope 2000, 15.)

David Cope on tehnyt uraa uurtavaa työtä jo pitkään algoritmisen säveltämisen parissa. Siksi tässä opinnäytetyössä keskitytään suurimmaksi osaksi hänen työnsä tuloksiin. Käytän edellä mainittua kirjaa suurimmaksi osaksi lähdemateriaalina. Mainittakoon vielä, että David Copen kirjoittaman trilogian ensimmäinen osa, "Computers and Musical Style", liittyy tietokoneen vastuulle annetulle musiikkianalyysille. Trilogian toinen osa, "Experiments in Musical Intelligence", keskittyy kuvaamaan David Copen kirjoittamaa Emi-tietokoneohjelmaa, joka luo uutta musiikkia lähtöaineistosta tehdyn analyysin perusteella.

1.3 Algoritmisen säveltämisen määritelmä tulevaisuudessa?

Tekoälystä on puhuttu viime vuosina paljon. Mielestäni tekoälyn avulla voidaan tehdä suuria harppauksia eteenpäin – myös algoritmisen säveltämisen saralla. Tulevaisuudessa algoritmisen säveltämisen määritelmä voi mahdollisesti muuttua ja esimerkiksi jakautua useampaan osaan. Esimerkiksi pelkästään tietokoneen "itsensä" tekemää säveltämistä voitaisiin kutsua jollakin muulla nimellä kuin algoritmisen säveltäminen. Tällä hetkellä ihminen on enemmän tai vähemmän mukana tietokoneen sävellysprosessissa. Ei kuitenkaan niin, että ihmisen myötävaikutusta ei tarvittaisi, jotta tietokone säveltäisi "itse".

Tekoälyn merkitys on lisääntynyt huimasti teollisuudessa ja yksi näistä teollisuudenaloista on musiikkiteollisuus. Seuraavassa luvussa, eli luvussa 2, käsitellään tekoälyä ja sen viime aikaista "vallankumousta" ennen kuin siirrytään tarkastelemaan algoritmista säveltämistä tarkemmin, luku 3.

2 TEKÖÄLYN VALLANKUMOUS

Algoritminen säveltäminen voi ottaa suuria harppauksia (lähi)tulevaisuudessa tekoälyyn liittyvien tekniikoiden kehittyessä yhä pidemmälle. Monilla aloilla on jo nähty, miten tekoäly on muuttanut työskentelytapoja: monia työpaikkoja on hävinnyt tekoälyn ollessa parempi ja tehokkaampi kuin ihminen vastaavassa tehtävässä. Myös musiikkialalla on nähtävissä tekoälysovellusten merkityksen kasvu.

2.1 Tekoäly on todellisuutta jo nyt

Toimittaja Samuli Kotilainen kirjoittaa Tietoviikko-lehdessä syväoppimisen, tekoälyn merkittävän läpimurron, olevan ihmiskunnan suurin keksintö sitten tieteellisen menetelmän. Hänen mukaansa syväoppimista otetaan käyttöön yhä useammalla alalla ja pian sen käyttö on saavuttamassa työmarkkinat. (Kotilainen, 2017, 18.)

Kotilainen kertoo, että esimerkiksi puheentunnistamista tekevät järjestelmät olivat vuosia naurun aiheena, koska ne eivät toimineet kunnolla. Tässä tapahtui kuitenkin äkillinen muutos ja nykyään esimerkiksi älypuhelin voi ymmärtää yllättävän hyvin jopa suomen kieltä. Facebook esitteli myös vuonna 2016 ominaisuuden, mikä pystyy Facebookin mukaan tunnistamaan kuvasta eri ihmisiä noin 97% tarkkuudella. Autojen kyky ajaa itsenäisesti kehittyy koko ajan. (Kotilainen, 2017, 20.)

Andrew Ng (2017) toteaa ”The State of Artificial Intelligence” -luennossaan, että tekoäly on ’nykyajan sähköä’ - sitä käytetään kohta kaikkialla. Hänen mukaansa äänihaut ovat jo niin hyviä, että ne saavat selvää käyttäjän kysymyksestä, joten ihmiset käyttävät äänihakua enemmän kuin aiemmin ja 10% hauista tehtiin vuonna 2017 äänen avulla. Lisäksi nykyään dataa tallennetaan yhä enenevässä määrin tietokoneille (tietokantoihin). Jos halutaan hyvää tehokkuutta ja oikeaa toimintaa tietokoneelta jossain tehtävässä, pitää tehdä vain iso neuroverkko ja antaa sille laadukasta dataa millä harjoitella. (Ng 2017) Vastaus siihen miksi tekoälysovellukset ovat viime aikoina parantaneet toimintaansa löytyy syväoppimisesta.

2.1.1 Syväoppiminen

Kotilainen (2017) mainitsee tekoälyn parantumisen taustaksi merkittävän tieteellisen ja teknisen läpimurron. Tämä läpimurto on nimeltään 'deep learning' eli suomeksi syväoppiminen. Kotilainen kirjoittaa, että vihdoinkin tietokoneet voivat oppia itsenäisesti ja tämä taas on mahdollistanut tekoälyn tekemät hurjat harppaukset viime vuosina. Lähivuosien vaarana voi olla, että tekoäly paranee niin paljon tehokkuudessa ja toiminnassaan, että jopa jotkut teollisuudenalat voivat olla vaarassa siinä mielessä, että aiemmin ihmisten tekemiä töitä annetaan tekoälyn tehtäväksi. Kotilainen on haastatellut muun muassa tunnettua teknologiasijoittajaa Steve Jurvetsonia vuonna 2016 ja hän totesi seuraavaa: ”*Syväoppimisen avulla tietokone voidaan opettaa tekemään mitä tahansa ja se tulee mullistamaan maailmamme nopeammin kuin kukaan uskoo.*” (Kotilainen, 2017, 20.)

Syväoppimisen Kotilainen kertoo olevan tekniikkaa, joka yrittää jäljitellä ihmisen aivojen toimintaa. Vielä viitisen vuotta sitten samaa tarkoittavana asiana käytettiin sanaa neuroverkko. Nimi kuitenkin vaihdettiin, koska neuroverkot saivat huonon maineen, kun niille asetettuja suuria toiveita ja odotuksia ei pystytty täyttämään. Esimerkkinä Kotilaisen haastattelema TTY:n professori Joni Kämäräinen toteaa, että harkitsi vuonna 2010 neuroverkkoluentojen lopettamista. (Kotilainen 2017, 20.)

Kotilainen kertoo, että tekoälyteknologian tilanne muuttui hetkessä, parissa vuodessa, kun neuroverkkojen tutkimisessa tehtiin läpimurtoja samalla kun tietokoneiden laskentatehot kasvoivat. Vuonna 2012 syväoppimista käyttävä järjestelmä tunnisti kuvia kaksi kertaa tarkemmin ImageNet-kuvantunnistuskilpailuissa kuin toiseksi tullut tekoälyjärjestelmä. Lisäksi syväoppimista käyttävä tekoäly on käytössä DLA Piperissä, joka on maailman toiseksi suurin lakiyhtiö. Lakiyhtiö on antanut koneoppivan järjestelmän tehtäväksi käydä suuria määriä sopimusaineistoa läpi ja löytää sieltä keskeisiä lausekkeita ja mahdollisesti tunnistaa aiemmin analysoidun aineiston perusteella riskejä: tekoäly voi jopa yrittää tehdä johtopäätöksiä sopimuksen vaikutuksista. (Kotilainen 2017, 20-21.)

Edellä mainittuja esimerkkejä lukiessa voi tulla pelko, että tekoäly voisi korvata myös säveltäjän tekemän työn musiikin parissa jossain vaiheessa. Jos kerran

tekoäly pystyy tekemään johtopäätöksiä lakifirman kiemuraisista sopimuspykälästä, niin miksi se ei pystyisi myös jossain vaiheessa säveltämään mestariteoksia – kuten Mozart ja Beethoven aikanaan. Voisiko syväoppiminen olla sellainen asia, joka mahdollistaisi tietokoneen sävellystaitojen huiman kasvattamisen?

Niklas Storås on haastatellut Timo Honkelaa, joka on tekoälyn tutkija. Storås kertoo 30-vuotta tekoälyä tutkineen Honkelan todenneen, että tekoälyä on aikaisemmin kehitetty kielellisiin asioihin liittyen (kuten esimerkiksi puheentunnistus). Tulevaisuudessa 'keinoäly' pystyy Honkelan mukaan ymmärtämään myös esimerkiksi näkö- ja kuulohavaintoja. Tämä johtaa Honkelan mukaan siihen, että koneet saavat yhä enemmän tietoa ja niiden ymmärrys tulee enemmän ihmisen kaltaiseksi. (Storås 2017, 26)

2.1.2 Mitä syväoppiminen oikeastaan on?

Kotilainen toteaa neuroverkkojen olevan valtava määrä niin sanottuja neuroneita, jotka ratkaisevat tehtäviä. Jokaisella neuronilla on joku yksinkertainen tehtävä – ei yritetä tehdä liian paljon yhdellä kertaa, vaan käytetään ohjelmoinnista tuttua hajoita-ja-hallitse-periaatetta, jotta saadaan muodostettua pienistä osista valmis kokonaisuus. Ratkaisua tarvitseva ongelma jaetaan pieniin osiin, jolloin kokonaisuutta on helpompi hallita – ratkaistaan jokainen pieni yksittäinen osa ongelmasta, jolloin saadaan lopulta kokonaisuus ratkaistua. Neuroverkko voi koostua monesta kerroksesta – puhutaan neuroverkkojen syvyydestä. Kun neuroverkkoa syvennetään, se voi oppia tekemään yhä monimutkaisempia asioita. (Kotilainen 2017, 22.)

Kotilainen mainitsee hyvänä esimerkkinä kuvantunnistuksen. Sen sijaan, että yritetään ohjelmallisesti luoda säännöt sille minkälainen mikäkin kissa on, käytetäänkin syväoppimista, jotta saadaan tietokone tunnistamaan mikä kissa tietokoneen tutkittavaksi annetussa kuvassa on. Idea on siis se, että annetaan tietokoneen tutkittavaksi iso määrä analysoitavaa aineistoa, jossa on kerrottu mikä eläin – oli se sitten esimerkiksi kissa tai koira – kuvassa on. Tietokone alkaa opetella neuroverkkoa apuna käyttäen samaansa aineistoa ja muodostaa yhä syvempiä ja tarkempia tasoja neuroverkkoon, jotta saadaan tunnistettua esimerkiksi 90% tarkkuudella onko kuvassa kissa vai ei. Tietokoneen tekemä analysointi voi viedä

paljonkin aikaa, jopa viikkoja, riippuen muun muassa siitä, mikä tehtävä on kyseessä ja paljonko aineistoa pitää analysoida. (Kotilainen 2017, 22.)

Mitä enemmän ja tarkempaa dataa tekoälyjärjestelmälle annetaan, sitä taitavamaksi järjestelmä sille annetussa tehtävässä tulee. Syväoppimista on käytetty muun muassa perinteisesti tekoälylle vaikeissa tehtävissä, kuten esineiden poimimisessa. Yhdysvalloissa MIT-yliopistossa laitettiin tietokone itse oppimaan, putosiko joku esine, kun se yritti nostaa sitä. Robotti jätettiin yksin tekemään tätä oppimista. Muutaman sadan tunnin harjoittelun ja oppimisen tuloksena järjestelmä pystyi jo aika hyvin nostamaan esineen pudottamatta sitä. (Kotilainen 2017, 22.)

Kotilainen vahvistaa, että syväoppimisen aika on nyt ja tekoälyn kehittäminen on käynnissä monilla aloilla. Lähitulevaisuudessa voidaan nähdä uusia mullistavia sovelluksia esimerkiksi aloilla, joissa perinteisesti käydään paljon kirjoitettua tekstiä läpi. Tekoälyn kehitys on jo niin pitkällä, että on olemassa järjestelmä, joka pystyy annetun valokuvan ja Google Earthin avulla kertomaan paikan koordinaatit maapallolla. (Kotilainen 2017, 22.)

2.1.3 Syväoppimiseen liittyviä ongelmia

Kotilainen kirjoittaa artikkelissaan, että tekoäly ei voi esimerkiksi ymmärtää tilannetta, jossa ihminen taluttaa tien yli koiraa – pitäisikö autoa ohjaavan tekoälyn pysähtyä vai jatkaa ajamista eteenpäin? Sama ongelma on kyseessä tekoälyn havaitessa edessä valkeaa massaa (lumisade) ja yrittää sitten päätellä onko kyseessä päälle ajava auto vai jotain muuta. (Kotilainen 2017, 22.)

Ongelma syväoppimisessa on esimerkiksi segmentointi. Syväoppimisen seurauksena saatuja tietoja ei voida yhdistää, koska ei ole olemassa kokonaisrakennetta – tekoäly ei siis osaa yhdistää asioita kokonaisuuksiksi. Kun kokonaisuuden hahmottaminen puuttuu, on tekoälyllä vaikeuksia hahmottaa objekteja. Lisäksi ihminen pystyy oppimaan lukemalla itsekseen kirjoja tai katselemalla YouTube-videoita, jotka liittyvät hänen kiinnostuksen aiheeseensa. Tekoäly ei tällaiseen oppimisen tasoon vielä pysty syväoppimisen tarjoamin keinoin. (Kotilainen 2017, 22.)

2.2 Tekoäly käytössä tänä päivänä

Cope toteaa, että ei niin kaukaisessa lähitulevaisuudessa algoritmista säveltämisestä tulee enemmän tai vähemmän 'normaali' lähestymistapa musiikin säveltämiseen. Monet säveltäjät ymmärtävät algoritmisen säveltämisen taustalla olevat ideat ja pitävät tietokoneen käyttämistä säveltämisen apukeinona hyödyllisenä. Copen mukaan monet säveltäjät käyttävät formaaleja algoritmeja eivätkä oikeastaan enää ajattele sen olevan mitenkään kummallista. Nämä säveltäjät eivät kuitenkaan mainitse asiasta, koska musiikkiyhteisön sisältä voi tulla herkästi negatiivista palautetta. Cope itse on saanut tämän negatiivisen palautteen kokea Emin tekemien sävellysten myötä. (Muscutt 2007)

Tulevia musiikin ammattilaisia voi kiinnostaa seuraava kysymys: "Onko mahdollista, että jossakin vaiheessa tekoäly voi säveltää yhtä hyvin kuin ihminen?". Monella alalla voi tulla vastaan painetta siirtää osa tai kaikki nykyään ihmisen tekemistä tehtävistä tietokoneen vastuulle. Onko siis aihetta olla huolissaan siitä, että tietokone veisi myös musiikin säveltäjiltä työt? Palaan näihin kysymyksiin tässä työssä myöhemmin tulevassa Pohdinta-luvussa.

Kotilainen pohtii artikkelissaan, onko tekoälyn käyttämisestä enemmän haittaa kuin hyötyä. Mitä muutoksia yhteiskunnassa pitää tehdä, jos syväoppimista käyttävä järjestelmä syrjäyttää ihmisen tekemiä töitä? Hänen mukaansa esimerkiksi Yhdysvalloissa pelätään robottiautojen vievän rekkakuski- ja taksinkuljettajien työt muutamien vuosien päästä. Yhteiskunnallisia ongelmia syntyy, jos tekoäly alkaa viedä parissa kymmenessä seuraavassa vuodessa ihmisiltä yhä enemmän ja enemmän työpaikkoja – esimerkiksi jopa 50% nykyisistä työpaikoista. Kuitenkin tekoälyssä ja syväoppimisessä on paljon hyviä puolia, kuten rutiininomaisten ja tylsien tehtävien poistaminen ihmisten harteilta. Tekoäly voidaan nähdä sekä uhkana että mahdollisuutena. Tällä hetkellä paras ratkaisu näyttäisi olevan, että työ tehdään ihmisen ja tekoälyn yhteistyönä. (Kotilainen 2017, 25.)

2.3 Tekoäly ja sen suhde algoritmiseen säveltämiseen?

Tekoäly voi tulevaisuudessa ohjalla yhä enemmän algoritmista säveltämistä: auttaa säveltäjää hänen sävellystyössään, esimerkiksi tarjoamalla nykyisiä sovelluksia paremmin jatkoehdotuksia sävellyksen osalle. Voidaan sanoa, että jo tällä hetkellä erilaisissa algoritmisen säveltämisen sovelluksissa käytetään tekoälyä hyödyksi.

Tarkoitukseni ei ole käydä tekoälytekniikoita läpi tässä opinnäytetyössä. Enemmän minua kiinnostaa algoritmiseen säveltämiseen liittyvät prosessit: mistä osista algoritminen säveltäminen koostuu ja mitä käyttäjän, eli tässä tapauksessa säveltäjän, tulisi tietää prosessin kulusta ja tehdä prosessin aikana. Seuraavassa luvussa tutustutaan siihen mitä algoritmisessa säveltämisessä tapahtuu. Myös David Copen ohjelmien käyttämää algoritmisen säveltämisen prosessia käydään läpi. Lisäksi käydään läpi myös vastaavanlaista ideaa popmusiikin säveltämisessä.

3 MITÄ ALGORITMISESSA SÄVELTÄMISESSÄ TAPAHTUU

Algoritmisessa säveltämisessä on erilaisia vaiheita riippuen siitä, millaisia menetelmiä sävellyksen tekemisessä käytetään. Olen valinnut tarkasteluun yhden yleisesti käytetyn lähestymistavan.

3.1 ”Kerää-analysoi-luo” -lähestymistapa

Seuraavissa kappaleissa kerrotaan yhden lähestymistavan vaiheista ottamatta kantaa siihen, mitä tekniikkoja tietokoneella koodatut algoritmit käyttävät eri vaiheissa. Tämä lähestymistapa on kolmiosainen: 1) datan kerääminen, 2) datan analysoiminen, 3) sävellyksen tekeminen.

3.1.1 Datan kerääminen

Jotta saataisiin luotua sääntöjä ja malleja sävellystä varten, pitää olla jotain lähtödataa, jota tietokone voi tutkia ja sitten käyttää hyödyksi samaansa tietoa. Tämä datamassa, eli joku määrä lähtöaineistoa, tarjoaa koneelle analysoitavaa ja, riippuen algoritmien tehokkuudesta, tietokone pystyy tekemänsä datan analysoinnin perusteella luomaan sääntöjä.

David Cope toteaa, että hän hankki 70 kappaletta tärkeinä pitamiensä orkesteriteosten osia MIDI-tiedostoina internetistä Emi-ohjelmaansa varten. Hän valitsi tietokantaansa sellaisia osia, joista olisi eniten hyötyä. Näillä osilla oli muun muassa sama tempo, moodi ja tahtilaji. Tiedostojen lataamisen jälkeen Cope joutui tarkastamaan ne käsin: hän siirsi ne ensin nuotinnusohjelmaan ja korjasi sitten havaitsemiaan virheitä. Copen mukaan on todella tärkeää, että analysoitava tietokanta on mahdollisimman eheä ja sisältää oikeaa dataa. Virheiden etsiminen ja korjaaminen oli yksi työläimmistä vaiheista. Hän valitsi lopulta 20 kappaletta alkuperäisistä 70 kappaleesta. Noissa 50 kappaleessa, joita ei valittu mukaan, oli liian uniikkeja kohtia, tunnistettavia piirteitä tai liian paljon korjattavia virheitä. Cope käytti omien sanojensa mukaan yli 50 tuntia muun muassa virheiden korjaamiseen ja tietojen syöttämiseen tietokantaan – silloinkin hän oli vasta työn alkuvaiheessa! Yksi syy siihen, miksi Cope ei enää käytä Emiä tonaalisen musiikin säveltämiseen, on juuri tämä prosessi tietokannan valmiiksi saattamiseen Emin käyttöä varten – se on yksinkertaisesti liian työläs ja aikaa vievä prosessi. Ensimmäisen Bachin koraalien tietokannan tekeminen kesti Copelta monta kuukautta,

mutta työn valmistuttua hän pystyi luomaan minuutissa satoja uusia koraaliteoksia. Hän kuunteli innoissaan yhden ja toisen, kunnes löysi miellyttävän kuulaisen koraaliteoksen. Hänen mielestään kaikki aika, mikä kului tietokannan luomiseen, oli vaivan arvosta. (Muscutt 2007)

3.1.2 Datan analysointi

Datan keräämisen jälkeen se pitää analysoida, jotta päästään haluttuun lopputulokseen – tässä tapauksessa tietokoneen tekemään sävellykseen. Datan analysoinnin aikana tietokone yrittää löytää säännönmukaisuuksia ja eroavaisuuksia tutkittavasta datamassasta. Sen jälkeen tietokone alkaa muodostaa sääntöjä esimerkiksi kyseisen musiikkityylin rakenteesta.

3.1.3 Sävellyksen tekeminen

Datan analysoinnin jälkeen on vuorossa uuden sävellyksen tekeminen. Tässä kohtaa tietokone pyrkii löytämiensä sääntöjen perusteella luomaan uuden kappaleen. Lopputuloksena syntyneen sävellyksen (ihmisten kokema/tuntema) laatu riippuu paljon esimerkiksi 1) tietokoneelle annetun lähtödatan laadukkuudesta (onko tietokoneen käyttämässä lähtödatassa paljon vai vähän virheitä, ovatko lähtödatana käytetyt kappaleet itsessään laadukkaita vai eivät), 2) tietokoneen käyttämästä algoritmista (algoritmin tehokkuus, sen toimivuus ja oikeellisuus) ja 3) datan analysoinnin perusteella tehdyistä säännöistä (onko tietokone osannut löytää oikeita sääntöjä, ovatko säännöt todellisia vai tietokoneen luomia harhoja kyseisistä kappaleista/musiikkityylistä). Huomautettakoon tässä vielä, että en löytänyt vastausta siihen, miten saadaan selville ovatko tietokoneen tekemät säännöt todellisia vai harhoja.

3.2 Esimerkkitapaus popmusiikista

Seuraavaksi esitellään yhteenveto eräästä opinnäytetyöstä (Lorentzon ynnä muut, 2015) liittyen popmusiikkiin ja algoritmiseen säveltämiseen. Siinä kerrotaan, miten tietokone laitettiin säveltämään popmusiikkia.

3.2.1 Popmusiikkia tietokoneen säveltämänä

Lorentzon ynnä muut (2015) käsittelee algoritmista säveltämistä popmusiikin näkökulmasta. Lähtökohdaksi on otettu koneoppimiseen liittyvien keinojen hyödyn-

täminen, kuten dynaamiset Bayesilaiset neuroverkot ja Markovin ketjut, jotta tietokone saadaan säveltämään popmusiikkia. Tutkimus on mielenkiitoinen jo senkin vuoksi, että aika ajoin kuulee pohdintaa voisiko tietokone säveltää 'ihmisen lailla'. Tietokoneen sävellyksistä, oli sitten kyseessä klassinen musiikki tai vaikkapa popmusiikki, on mielenkiintoista tutkia, minkälainen kappale on: mitä sointuja ja rytmejä on käytetty, millainen melodia on, mikä kappaleen rakenne on ja niin edelleen.

Tietokoneiden kehittyttyä yhä nopeammiksi ja enemmän muistia sisältäviksi on 2000-luvulla tehty enemmän tutkimusta algoritmiseen säveltämiseen liittyen kuin sitä ennen: muun muassa jazz-soolojen säveltäminen tietokoneen toimesta sille annetun säännöstön/kieliopin mukaan on ollut tutkimuksen kohteena. (Lorentzon ynnä muut, 2015)

Jotta tietokone voisi oppia popmusiikin sääntöjä, tarvitaan ensin jonkinlainen aineisto tietokoneen analysointia varten. Opinnäytetyön tekijät saivat käyttöönsä Hooktheoryltä MIDI-tietokannan, jossa oli tarpeeksi pop-kappaleita data-analyysin tarpeisiin. Hooktheory.com on sivusto musiikin opiskeluun. Sieltä löytyy muun muassa ohjelmistoja, kirjoja, videoita ja artikkeleita. Jos olisi käytetty esimerkiksi MP3-tietokantaa, olisi pitänyt käyttää tietokoneen kannan data-analysoinnissa muun muassa frekvenssianalyysia, jotta olisi saatu selville mitä sointuja eri kohdissa kappaletta on käytetty. MIDI-tietokannassa on se hyvä puoli, että esimerkiksi sointujen saaminen selville on helppoa riippumatta siitä missä kohtaa kappaletta mennään, koska MIDI datassa kerrotaan muun muassa nuotin syttymis- ja sammumisaika.

Opinnäytetyön tekijät käyttivät seuraavaa algoritmia tietokoneen säveltämän kappaleen tekemiseen:

1. Analysoidaan olemassa olevia pop-kappaleita kolmella eri tavalla: 1) mitä sointuja on käytetty, 2) miten melodia on rakennettu ja 3) mitä rakenteita kappaleessa on käytetty.
2. Annetaan tietokoneen harjoitella datamassalla. Näin saadaan aikaan sääntöjä esimerkiksi siitä, miten sointuja on käytetty, miten melodia on sävelletty ja millainen kappaleen rakenne on.

3. Pyydetään tietokonetta säveltämään oppimansa ja havaitsemansa perusteella omia tyylin mukaisia kappaleita.

Kun tietokone oli säveltänyt noin 250 kappaletta, valitsivat opinnäytetyön tekijät niistä viisi sellaista, mitkä muistuttivat heidän mielestään eniten ihmisen säveltämää pop-kappaletta. Tämän jälkeen he valitsivat oikeiden ihmisten säveltämiä (vähemmän tunnettuja) pop-kappaleita satunnaisesti Hooktheory:n MIDI-tietokannasta. Lopullisessa testiaineistoissa oli viisi tietokoneen säveltämää kappaletta ja neljä ihmisten säveltämää kappaletta. Tekijät järjestivät sitten kuuntelijatutkimuksen. Tutkimuksessa ihmisiä pyydettiin kuuntelemaan nämä yhdeksän kappaletta ja sitten valitsemaan jokaisen kappaleen kohdalla kolmesta eri vaihtoehdosta (ihmisen tekemä, tietokoneen tekemä, olen kuullut tämän kappaleen aikaisemmin) heidän mielestään sopivin.

Tilastomatematiikkaa käyttäen saatiin lopputuloksena seuraavaa. Tietokoneen säveltämissä kappaleissa oli jotain sellaista mikä sai ihmiset 'tunnistamaan', että kyseessä on luultavasti tietokoneen tekemä kappale. Jos keskiarvo olisi ollut lähellä 50%, niin se tarkoittaisi opinnäytetyön tekijöiden mukaan sitä, että olisi lähes mahdotonta tietää onko kappaleen tehnyt ihminen vai tietokone. Heidän tekemässään kuuntelijatutkimuksessa saatiin keskiarvoksi 58%.

Jotta yllä kerrottu kuuntelijatutkimuksen lopputulos tulisi lukijalle paremmin selväksi, niin kerrataan asia. Saatu tulos tarkoittaa sitä, että kuuntelijat pystyivät tunnistamaan tietokoneen tekemiä pop-kappaleita: niissä oli jotain sellaista mitä ihmissäveltäjät eivät olisi tehneet. 58% on sellainen tulos, että vaikka se tuntuu pieneltä erolta, niin sitä se ei todellisuudessa ole. Yksi selittävä tekijä siinä että ihmiset pystyivät tunnistamaan tietokoneen tekemiä pop-kappaleita on se, että tietokoneen käyttämä algoritmisen säveltämisen algoritmi tarkasteli vain melodiaa, sointuja ja kappaleen (laajempaa) rakennetta. Esimerkiksi David Copen Emi-ohjelma on mitä luultavammin paljon kehittyneempi ohjelma, koska David Cope on käyttänyt monta kymmentä vuotta ohjelmansa kehittämiseen – kun taas ruotsalaisten käyttämä ohjelma on varsin yksinkertainen (opinnäytetyön lukemisen perusteella). David Copen omien kuuntelijatestien mukaan ihmiset eivät pysty, edes asiantuntijat, erottamaan hänen ohjelmansa parhaimpia sävellyksiä tietokoneen tekemiksi.

Lorentzon ynnä muut kertovat, että käytetyissä koneoppimisen menetelmissä tuli vastaan tietokoneen muistin riittäminen – olisi tarvittu enemmän muistia, jotta olisi saatu tietokone oppimaan ja säveltämään paremmin popmusiikkia. Tietokoneen säveltämät kappaleet eivät myöskään tuntuneet tarpeeksi omilta kokonaisuuksiltaan – tietokoneen tekemän kappaleen kokonaisrakenne ei ollut tarpeeksi ehyt. Popmusiikki on verrattuna klassiseen musiikkiin yleensä yksinkertaisempaa: esimerkiksi samat sointukulut toistuvat usein eri kappaleissa popmusiikissa.

Tekijät ehdottavat lopuksi, että säveltäjät voisivat käyttää tietokoneen säveltämiä kappaleita inspiraation lähteenä ja valita tietokoneen tekemistä sävellyksistä niitä mitkä kuulostavat heidän korvaansa hyviltä ja lähteä sitten parantelemaan niitä.

3.3 Tietokone säveltää barokkimusiikkia YouTube-video

YouTubesta löytyy muun muassa video, jossa tietokone laitetaan säveltämään barokkimusiikkia Bachin tyylillä – tai toivotaan, että tietokone osaisi säveltää uutta Bachin tyylistä musiikkia. Lähtödatana syötetään joitakin Bachin teoksia ja sitten annetaan tietokoneen harjoitella hetken aikaa – ja sitten vähän pidemmän aikaa. Lopputulos paranee aikaa myöden hieman, mutta lopulta videon tekijä toteaa, että vaikka harjoittelua jatkettaisiin vielä pidempään, niin käytetyllä algoritmilla tuskin päästään kovinkaan parempiin tuloksiin. Videon pääsi katsomaan maaliskuussa 2020 YouTubesta ja linkki on seuraava: https://www.youtube.com/watch?v=SacogDL_4U

Video kertoo yhdestä lähestymistavasta algoritmiseen säveltämiseen liittyen. Videon pituus on myös aika lyhyt, hieman yli 18 minuuttia. Mielestäni video kuvaa hyvin algoritmista säveltämistä: lähtödataa syötetään tietokoneelle, tietokone yrittää säveltää samantyylistä musiikkia kuin lähtödata on ja ihminen kuuntelee mitä lopputuloksena saatiin aikaiseksi.

3.4 Käytettyjä menetelmiä ja ohjelmistoja?

Tässä luvussa on käsitelty yksi lähestymistapa algoritmiseen säveltämiseen liittyen: datan kerääminen, datan analysoiminen, uuden sävellyksen tekeminen. Lukijaa saattaa kiinnostaa mitä menetelmiä ja tekniikoita on käytettävissä. Lisäksi lukijaa luultavasti kiinnostaa, mitä ohjelmistoja on olemassa algoritmiseen säveltämiseen liittyen.

Seuraavassa luvussa, luvussa 4, käydään pintapuolisesti läpi joitakin tekniikoita/menetelmiä, mitä voidaan käyttää algoritmisen säveltämisen mahdollistamiseksi tietokoneella. Tarkoitus on listata lyhyesti joitakin tekniikoita ja ohjelmistoja – ei käydä yksityiskohtaisesti läpi esimerkiksi mitä Markovin ketjuilla tarkoitetaan. Lukijan on hyvä saada se käsitys, että algoritmisessa säveltämisessä on paljon kyse matematiikasta ja sen hyödyntämisestä niin, että tietokone saadaan avustamaan käyttäjää sävellyksen tekemisessä.

4 ALGORITMISEN SÄVELTÄMISEN MENETELMIÄ JA OHJELMISTOJA

Millaisia tekniikoita ja menetelmiä algoritmisessa säveltämisessä käytetään? Tässä luvussa tarkastellaan lyhyesti, millaisia menetelmiä on käytössä ja miten ihminen voi vaikuttaa sävellysprosessin kulkuun. Lopuksi listataan joitakin ohjelmistoja.

Mainittakoon vielä, että viime vuosien tekoälytekniikoiden nopean kehittymisen takia on varmasti olemassa tässä listattuja tekniikoita uudempiakin tekniikoita. Niitä ei kuitenkaan ole nyt otettu tarkasteluun, koska tarkoitus on vaan mainita joitakin algoritmisessa säveltämisessä käytettyjä (tekoäly)tekniikoita. Nykyaikaisiin algoritmisen säveltämisen tekniikoihin liittyen on lukijan etsittävä itse esimerkiksi sopivia kirjoja ja artikkeleita aiheeseen liittyen. On hyvä myös huomata, että vaikka käyttämäni esimerkit, kuten Markovin ketjut, ovat aika vanhoja keksintöjä, niin niiden hyödyllisyys esimerkiksi algoritmisessa säveltämisessä ei katoa mihinkään – mainitut tekniikat ovat edelleen oleellisia, vaikka uusia (tekoäly)tekniikoita kehitetään.

Viime vuosina monet isot yritykset, kuten Sony ja Google, ovat panostaneet algoritmisen säveltämisen tutkimiseen ja tekniikoiden kehittämiseen. Esimerkiksi Googlen Magenta-projekti, <https://research.google/teams/brain/magenta/>. Magenta keskittyy koneoppimisen tekniikoihin taiteellisen oppimisprosessin näkökulmasta. On yleensä hyvin todennäköistä, että isot yritykset eivät halua avata käyttämiään (tekoäly)tekniikoita muun muassa yrityssalaisuuksien takia. Magenta-projekti on nähtävästi kuitenkin poikkeus tähän sääntöön. Kyseessä on nimittäin avoimen lähdekoodin projekti. Avoimella lähdekoodilla tarkoitetaan yksinkertaisesti selitettynä sitä, että kuka tahansa voi saada ohjelman lähdekoodin haltuunsa ja muokata sitä halunsa mukaan. Magenta-projektin tarkoitus on hyödyntää syväoppimisen menetelmiä esimerkiksi uuden musiikin ja kuvien tekemiseen. Magentan kotisivuilta, <https://magenta.tensorflow.org/>, löytyy lisää tietoa aiheesta kiinnostuneille.

4.1 Lähestymistapoja ja vaikuttamiskeinoja

Länsimainen nuottikirjoitus käyttää hierarkkista rakennetta kuten nuotteja ja fraaseja. Musiikkia voidaan 'ohjailla' esimerkiksi toistoilla ja hypyillä kuten *da capo al*

fine. Musiikkiin liittyvät formaalit kieliopit perustuvat näiden musiikin osatekijöiden mallintamiseen. Musiikki voidaan nähdä toinen toistaan seuraavina tapahtumina, joita sitten ohjaillaan käskyillä kuten hyppykäskyt ja toistokäskyt. (Järveläinen 2000)

Säveltämisessä käytetyt algoritmit voidaan jakaa käytännössä deterministisiin ja stokastisiin. Yhdessä kappaleessa voidaan käyttää useita algoritmeja ja lähestymistapoja, joten kuuntelijan voi olla lähes mahdotonta tunnistaa mitä menetelmää musiikin tekemiseen on käytetty. Jos algoritmi on kokonaan automatisoitu, ei algoritmin käyttäjän tarvitse kuin antaa jotain dataa analysoitavaksi ohjelmalle ja käynnistää algoritmin ajo tietokoneella. Käyttäjä voi vaikuttaa säveltämisprosessiin muokkaamalla ohjelmaa ja/tai sen parametreja. Käyttäjä kontrolloi algoritmia, joten lopputuloksena syntyvä sävellys, esimerkiksi sen laatu, lankeaa ihmiskäyttäjän harteille. Käyttäjä pystyy vaikuttamaan useaan eri sävellyksen osaan algoritmia muokkaamalla, kuten sävellyksen muoto ja jos mennään alimmille tasoille, niin päästään muokkaamaan jopa yksittäistä nuottia. Nykyajan tekniikalla pystytään muokkaamaan sävellystä reaaliajassa – siis samalla kun sävellys syntyy niin sitä voidaan muokata. (Järveläinen 2000)

Edwards kertoo 1900-luvun lopun tekniikkojen olevan hybridejä deterministisistä ja stokastisista menetelmistä. Tekoälyä tai kielitieteitä käyttäviä systeemejä ovat esimerkiksi Bol Processor -ohjelmisto ja Kerna Ebcioğlun CHORAL. Jotkut statistiikkaa käyttävät systeemit voivat olla esimerkiksi Markov-mallien laajennuksia. Näissä ongelmaksi muodostuu yleensä kuitenkin se, että tarvitaan paljon (ja laadukasta) lähtödataa, jotta systeemi saadaan oppimaan järkevällä tavalla. Edwards toteaa vielä, että nämä järjestelmät ovat tärkeitä tekoälyn tutkimukselle, kielitieteille ja ohjelmistotieteelle. Ne ovat kuitenkin aika rajallisia ominaisuuksiltaan sellaiselle säveltäjälle, joka tekee modernia (tai sitten omalla tyyllillään) musiikkia. (Edwards 2011)

4.1.1 Stokastisuus ja todennäköisyydet

Stokastisessa musiikissa on kyse todennäköisyyksistä. Päätöksiä tehdään satunnaislukujen perusteella. On siis mahdotonta ennustaa, millainen lopputulos syntyy. Stokastisia prosesseja on helppo käyttää ja niitä onkin käytetty säveltä-

misessä aina apukeinosta kokonaisen sävellyksen tekemiseen. Useimmissa menetelmissä käytetään todennäköisyystaulukkoa seuraavan nuotin valitsemiseen. Todennäköisyystaulukko kertoo kaikki mahdolliset vaihtoehdot ja jokaiselle vaihtoehdolle kerrotaan tämän toteutumisen todennäköisyys. Jos on N vaihtoehtoa ja kyseessä tasaisesti jakautunut todennäköisyys(kenttä), niin silloin yksittäisen vaihtoehdon toteutumisen todennäköisyys on $1/N$. Yleensä kuitenkin ei käytetä tasajakaumaa vaan halutaan, että joillakin nuoteilla on suurempi todennäköisyys esiintyä seuraavaksi. (Järveläinen 2000)

Ensimmäisenä säveltäjänä todennäköisyysjakaumia käytti Iannis Xenakis. Xenakis protestoi sarjallisen musiikin idealismia vastaan – hän piti jatkuvaa muutosta musiikillisen tiedon keskipisteenä. Xenakis alkoi käyttää samantyyllisiä stokastisia lakeja omassa musiikissaan: esimerkiksi sateen ropina kovalle pinnalle. (Järveläinen 2000)

4.1.2 Markovin ketjut

Markovin ketjut ovat diskreettejä systeemejä, joissa nykyisen tilan lopputulos riippuu sitä edeltäneiden tilojen lopputuloksista – eli nykyiseen tilaan päästään aikaisempien tilojen perusteella. Markovin ketjuista koostuvassa systeemissä on muistissa menneiden tilojen lopputulokset ja ne vaikuttavat tulevien tilojen lopputuloksiin. 0-tason Markov ketju on itse asiassa edellä mainittu todennäköisyysjakauma. 1-tason Markov ketju ottaa edellisen tason lopputuloksen huomioon. 2-tason Markov ketju ottaa 0-tason ja 1-tason lopputulokset huomioon. (Järveläinen 2000)

Yksi Markovin ketjujen ongelmista on se, että todennäköisyydet pitäisi luoda olemassa olevan musiikin perusteella. Tällöin saataisiin lopputulokseksi samaa musiikkityyliä oleva kappale. Markovin ketjuja käytetään pääasiassa melodioiden luomiseen. Jos Markovin ketjuja käytettäisiin myös esimerkiksi harmoniasuhteiden määrittämiseen, tarvittaisiin todella suuri todennäköisyysmatriisi (eli taulukko, missä on listattu kaikkien mahdollisten vaihtoehtojen todennäköisyys). (Järveläinen 2000)

4.1.3 'State machines' eli tilakoneet

Tilakone nimellä kutsutaan prosessia, jonka lopputulos riippuu systeemin sisäisestä tilasta ja siitä, mitä tietoa systeemiin on syötetty sisään. Musiikkiin liittyvissä ohjelmissa tilakoneilla mallinnetaan musiikillisia parametreja, kuten äänenkorkeutta ja sointiväriä. Tilakone määrittellään joukoksi tiloja ja näiden välisiä tilasiirtymiä, joihin sisään syötetty tieto vaikuttaa. Tilakoneita voidaan yhdistää, jolloin edellisen tilakoneen antama lopputulos syötetään seuraavaan tilakoneeseen sisään meneväksi tiedoksi. Systemi toimii itsenäisesti heti kun sille on syötetty lähtödataa. (Järveläinen 2000)

4.1.4 Muita menetelmiä

On olemassa myös muita algoritmisen säveltämisen menetelmiä kuten esimerkiksi 'Petri-verkot'. Näihin menetelmiin ei kuitenkaan tutustuta tässä työssä, vaan se jääköön lukijan oman mielenkiinnon varaan.

4.2 Joitakin ohjelmistoja

Seuraavaksi listataan joitakin ohjelmistoja algoritmiseen säveltämiseen liittyen. Ohjelmistot on valittu lähinnä sen perusteella mihin olen internetissä törmännyt, kun olen hakenut tietoa algoritmiseen säveltämiseen liittyen.

4.2.1 AIVA-ohjelmisto

AIVA, <https://www.aiva.ai/>, on tekoälyä hyväksikäyttävä ohjelmisto ja sen tarkoitus on auttaa säveltäjää luomisprosessissa. Tällä hetkellä AIVA voi luoda uutta musiikkia muun muassa seuraavissa tyyeissä: moderni elokuvamusiiikki, pop, rock, fantasia-musiikki, jazz ja tango.

TAULUKKO 1. AIVA-ohjelmistoon liittyviä opetusvideoita.

Video	Link
Creating EDM in 6 minutes w/ AIVA	https://www.youtube.com/watch?v=IJcmBmnyyY
How AI could compose a personalized soundtrack to your life Pierre Barreau	https://www.youtube.com/watch?v=wYb3Wimn01s

4.2.2 Flow Machines-ohjelmisto

Flow Machines, <https://www.flow-machines.com/>, on tekoäly- ja signaalin prosessointitekniikoita käyttävä musiikkia luova ohjelmisto. Sen tarkoitus on auttaa artistia luomaan uutta musiikkia. Käyttäjä pystyy pyytämään tekoälyä luomaan musiikkia useilla eri tyyleillä ja esimerkiksi melodia-, sointukulku- ja bassopohjan voi saada valmiina suoraan ohjelmalta. Käyttäjä voi sitten siirtää ohjelman luomat tiedostot esimerkiksi käyttämäänsä DAW-ohjelmistoon ja lisätä sitten itse lyriikoita ja miksata sekä masteroida kappaaleen. Flow Machines ei pysty luomaan itse kokonaista kappaletta, vaan sen tarkoitus on olla luovan artistin työtä helpottava ohjelmisto. Ohjelma voi tarjota käyttäjälleen inspiraation lähteitä muun muassa luomillaan melodiapohjilla.

4.3 YouTube opetusvideoita

Liitteessä 4 listataan joitakin opetusvideoita algoritmiseen säveltämiseen liittyen eli missä mennään algoritmisen säveltämisen hyödyntämisessä nykyaikana. Lisäksi on joitakin hyviä opetusvideoita liittyen tekoälyn kehittymiseen.

4.4 Historian havinaa - 2000-luvun alku

Cope listaa pienen joukon saatavilla olevia algoritmisen säveltämisen ohjelmistoja ”The Algorithmic Composer” -kirjassaan. Ymmärsin Copen tekstin niin, että nämä ohjelmistot ovat peräisin 1980-luvun lopusta ja 1990-luvun alusta ja ne eivät (ainakaan) silloin tarjonneet Copen tarvitsemia toimintoja. Tässä on lyhyt listaus niistä:

- Common Music, Patchwork ja Symbolic Composer.
 - Näistä Symbolic Composer on kaupallinen ja loput kaksi ilmaisia. Common Music on Stanfordin yliopiston kehittämä (oliopohjaista ohjelmointitapaa hyödyntävä) musiikin säveltämiseen tarkoitettu ohjelmistoalusta. Patchwork käyttää graafista käyttöliittymää kun taas Common Music on tekstieditorilla käytettävä. Symbolic Composer yrittää taas päästä eroon ’historian jäänteistä’ tulevasta musiikkinotaatiosta. Symbolic Composer käyttää kahta kielioppia määritellään musiikin osia ja instrumentteja.

Nämä ohjelmat tarjoavat Copen mielestä rikkaan kirjon erilaisia sävellystyötä tukevia työkaluja. Näillä työkaluilla ei Copen mukaan kuitenkaan saada uutta tietyn

tyylin mukaista musiikkia aikaiseksi. Sen takia Cope alkoikin itse tehdä omiin tarpeisiinsa sopivaa ohjelmaa – syntyi Emi. Tämä Copen tekemä ohjelmiston kehitys alkoi jo 1980-luvulla. (Cope 2000, 16.)

Seuraavassa luvussa, luvussa 5, kerrotaan David Copen tekemistä ohjelmistoista: Emistä ja Alicesta. Näiden ohjelmistojen pintapuolinen läpikäynti voi auttaa lukijaa ymmärtämään algoritmisen säveltämisen haasteita ja ratkaisukeinoja.

5 DAVID COPEN OHJELMISTOJA

David Cope on vuosien varrella tehnyt useampaa algoritmiseen säveltämiseen liittyvää ohjelmistoa. Tässä luvussa käsitellään niistä kahta: Emiä ja Alicea. Opin näytetyössäni on tarkoitus tarkastella algoritmista säveltämistä pintapuolisesti, mutta David Copen Emi ja Alice ohjelmat ovat omasta mielestäni hyvin tärkeitä virstanpylväitä algoritmisen säveltämisen historiassa. Tuon tässä luvussa esille joitakin suunnittelunäkökohtia David Copen itsensä kertomana. Lisäksi kerron yleiseltä kannalta David Copen ”The Algorithmic Composer” -kirjaan viitaten, miten Emi ja Alice toimivat. Asiasta kiinnostuneen kannattaa hankkia käsiinsä kyseinen kirja – se on hyvin aiheessa pysyvä ja yksityiskohtainen. David Cope osaa kertoa algoritmisesta sävellysprosessista loogisella ja selvällä tavalla.

5.1.1 David Copen Emi-ohjelma

Cope kertoo luoneensa Emi-ohjelman (ja sen alaohjelman Saran) omaan tarpeeseensa – hän ei löytänyt olemassa olevia ohjelmia, jotka voisivat luoda uutta halutun tyylin mukaista musiikkia tietokannasta löytyvien kappaleiden pohjalta. (Cope 2000, 19.)

Emi-ohjelma (Experiments in Musical intelligence) on säveltänyt muun muassa Bachin, Mozartin, Chopinin, Joplinin ja Gerschwinin tyyliä teoksia. Emi tutkii tietokannassaan olevia teoksia, analysoi ne etsiäkseen niin sanottuja ’allekirjoituksia’ (englanniksi ’signatures’) ja käyttää rekombinaatiota (eli ’leikkaa/liimaa’-menetelmää) säveltäessään uutta tietokannan tyyliä uskollista musiikkia. Emi on kirjoitettu Common LISP -ohjelmointikielellä.

Cope kertoo ensin syöttäneensä ohjelmalle sävellykseen liittyviä sääntöjä – eli miten hän, David Cope, on ymmärtänyt ja käyttänyt sääntöjä säveltäessään musiikkia. Tämä ei kuitenkaan johtanut Copea tyydyttäviin Emin tekemiin sävellyksiin – niistä löytyi sääntöjä noudattavaa yleistä ’massaa’ ja tyyliä uskollisia ratkaisuja oli vaikea löytää. Cope totesi, että on parempi, että hän ei syötä sääntöjä, vaan antaa Emi itse tutkia tietokantaa ja luoda siellä olevan datan (eli kappaleiden) perusteella säännöt musiikille. Copen mielestä on parempi, että annetaan lähtödatan analyysin jälkeen tietokoneen tehtäväksi sääntöjen luominen, sen sijaan että ihminen kirjoittaisi säännöt sisään ohjelmaan manuaalisesti eli käsin –

olisi todella työlästä kirjoittaa aina uuden tyylin tullessa esille sille säännöt tietokonetta varten.

Cope huomasi, että jokaisessa ihmisen tekemässä teoksessa on säännöt itsessään – olivat ne sitten tiedostettuna tehtyjä tai tiedostamattomana tehtyjä. Analyysin avulla näihin piilossa oleviin sääntöihin, kuten rakenne ja tyyllilliset hienoudet, päästäisiin käsiksi!

Cope käytti rekombinaatiota hyväkseen ja ohjelmoi Emin käyttämään tätä 'leikkaa/liimaa'-menetelmää. Hän teki myös uuden aliohjelman, minkä tarkoitus oli pilkkoa teokset iskuihin ja tutkia, miten äänet siirtyvät seuraavalta iskulta löytyville äänille. Tällä tavoin hän pystyi antamaan Emille keinoja äänenkuljetussääntöjen noudattamiseen. Emin säveltäessä tietokannasta etsitään vastaavia liikkeitä omaavia teoksia ja sitten yhdistellään näitä eri teoksien osia yhteen, jolloin tuloksena syntyy uusi teos. Tämä auttoi Copen mukaan jo huomattavasti tyylin muokausten sävellysten syntymisessä, ja mitä enemmän tietokannassa oli materiaalia, sitä enemmän vaihtoehtoja ohjelmalla oli valita mistä leikata ja liimata seuraava kohta.

Emin uusien teosten laatu kuitenkin vaihteli suuresti. Ongelmana oli muun muassa se, että syntyneissä sävellyksissä oli useita todella epätasapainossa olevia ja ei-tyypillisiä fraaseja – ei löytynyt mitään varsinaista musiikillista logiikkaa muuta kuin soinnusta toiseen siirtyminen. Lisäksi fraasien yhteen liittämisen Emi ei noudattanut mitään kaavaa. Emistä ei myöskään löytynyt logiikkaa/älyä musiikillisten ideoiden/aihioiden toistoon ja kehittelyyn.

Cope päätti muokata ohjelmaansa niin, että se ottaisi ylemmällä tasolla olevia musiikin rakenteita huomioon – hän ohjelmoi analyysimoduulin, jonka tehtävä oli muun muassa analysoida jokaiselta iskulta löytyvän harmonisen funktion merkitys ja verrata sitä edelliseen ja sitä seuraavaan iskuun. Hän teki kolmen soinnun analyysiyhdistelmiä ja tämä jo auttoi jonkin verran Emiä tekemään parempia sävellyksiä. Cope lisäsi myös kadenssien tunnistamiseen liittyvän osan analyysiohjelmaansa. Cope huomasi kuitenkin, että Emi vain toisti usein jo valmista Bachkoraalia tietokannasta, koska sääntöjen mukaan siitä löytyi paras seuraava isku, jonka Emi sitten leikkasi ja liimasi edellisen iskun jatkoksi. Tämän huomattuaan

Cope lisäsi ohjelmaan säännön, että se valitsee uuden kulkureitin aina kuin vain on mahdollista – eikä niin, että käytetään samaa koraalia jatkoa luotaessa. Cope havaitsi myös, että jos hän syötti tietokantaan muita teoksia kuin Bachin koraaleja, alkoi Emin tekemien uusien koraalien tyyli kärsiä. (Cope 2000, 19-22)

5.1.2 David Copen Alice-ohjelma

Cope toteaa, että Alice käyttää päättelyä (englanniksi ”inference”) ’leikkaa-liimaa’-menetelmän sijaan. Hän halusi, että Alice tuottaisi tietokannassa olevan musiikin perusteella uutta samantyylistä musiikkia. Tämä tapahtuu niin, että Alice analysoi tietokannastaan löytyvän musiikin, luo sitten säännöt musiikin tyyliin ja käyttää näitä sääntöjä uuden musiikin tekemisessä. Cope mainitsee, että tämä lähestymistapa algoritmiseen säveltämiseen tuottaa paljon parempia tuloksia kuin hänen aikaisemmin käyttämänsä Emi-ohjelma.

Sääntöjen luomisella on Copen mukaan kaksi suurta hyötyä: 1) sääntöjä voidaan muokata loogisella ja musiikillisella tavalla, 2) ohjelman käyttäjät saavat nähdä luodut säännöt ja tarvittaessa analysoida, verrata ja muokata niitä. Jotta tietokannassa oleva musiikki voidaan analysoida pitää se pilkkoa loogisiin osiin/ryhmiin: Emillä iskut, Alicella vaihteleva ryhmä koko riippuen ’tapahtumien’ syttymisistä ja sammumisista.

Ryhmät ovat tärkeässä roolissa, kun yritetään analysoida kappaleen tietyn kohdan musiikillista funktiota. Cope mainitsee esimerkiksi Bachin koraalien kolmisointuryhmittelyn tulevan tietokoneelle ongelmalliseksi, kun sen pitäisi päättää mitkä äänistä ovat harmonisia ja mitkä taas eivät ole. Ryhmäkoon määrittely on Copen mukaan ensimmäinen ratkaistava ongelma. Jos käyttää pientä ryhmäkoko, tulee syntyvästä musiikista hyvin erilaista kuin mitä tietokannasta löytyy. Jos taas käyttää hyvin suurta ryhmäkoko, tulee uudesta musiikista hyvin paljon samanlaista kuin mitä tietokannassa oleva – jopa liiankin samanlaista. Paras ryhmä koko olisi Copen mielestä sellainen, että syntyvä musiikki muistuttaa tietokannasta löytyvää musiikkia, mutta ei kuitenkaan ole tietokannasta löytyvän musiikin tapaista, vaan pikimminkin jotain tuoretta ja uutta – tietokannasta löytyvän musiikin tyylistä.

Atonaalisessa musiikissa tarvitaan sävelten ryhmittelyä loogisella ja toistettavalla tavalla. Atonaalista musiikkia voidaan esimerkiksi tutkia ja ryhmitellä melodisten linojen mukaan harmonioiden sijaan. Riippumatta tavasta, ryhmittelyn pitää olla loogista, samanlaisena pysyvää ja tarkoituksenmukaista.

Alicessa Cope on ryhmittelyn toteutuksessa käyttänyt 'tapahtumien' vertikaalista syttymistä ja sammumista. Cope toteaa, että tällä tavalla tyyli saadaan pidettyä samantapaisena, kun käytetään apuna 'allekirjoitusten' (englanniksi 'signatures') tunnistamista, meta-kaavamaisuuksia (englanniksi 'metapatterns'), korvamerkkejä (englanniksi 'earmarks') ja unifikaatiota. (Cope 2000, 67-73)

Tonaalisessa musiikissa Alice keskittyy sointuryhmien sekä äänenkuljetuksen analysointiin. Vaikka musiikkityylissä ei välttämättä aina ole äänenkuljetuksella suurta merkitystä, on Copen mukaan kuitenkin hyväksi, että Alice automaattisesti tekee äänenkuljetuksen analysoinnin. Sävelluokat (englanniksi 'pitch-class') on hyödyllistä analysoida – eli sävelen oktaavialalla ei ole merkitystä, kun se laiteetaan omaan sävelluokkaansa. Ongelmia voi syntyä, jos esimerkiksi kvarttiseksti-sointua käytetään heikolla tahtiosalla tai jos käytetään rinnakkaisia kvinttejä.

Ohjelman käyttäjä voi antaa negatiivisen palautteen ohjelman tarjoamalle musiikille, jolloin Alice yrittää uudelleen. Cope mainitsee, että asteikon asteet (englanniksi 'scale degrees') voidaan ottaa huomioon, jotta välttyään äänenkuljetuksessa liikkeiltä, jotka synnyttävät ongelmia. Esimerkiksi modaalisuus, tonaalisuus ja kokosävelasteikko eivät vaikuta Alicen tekemään analysointiin, vaan samaa sävelluokka-ajattelua voidaan käyttää modaalisen, tonaalisen ja atonaalisen musiikin analysointiin. (Cope 2000, 73-79)

Cope kertoo, että Alice käyttää päättelemiään sääntöjä sävellysprosessin lähtökohtana, kun taas edellä mainittu Emi-ohjelma hyödyntää tietokannasta löytyviä musiikillisia aineksia. Alicella on käytössään tekemänsä analysoinnin tuloksena saadut säveltasojoukot sekä äänenkuljetussäännöt. (Cope 2000, 88.)

Cope toteaa, että jos käyttäjä saisi syöttää säännöt itse, tulisi seuraavia ongelmia: 1) jokaista uutta tyyliä varten käyttäjän pitäisi syöttää säännöt, 2) käyttäjät

eivät saa kirjoitettua yleensä koko musiikkityyliä säännöiksi, vaan saavat muodostettua jonkinlaisen 'yleisen' tyylin mukaisen säännöstön, 3) käyttäjän syöttämien sääntöjen tekeminen vaatii paljon aikaa ja koodia, koska säännöt voivat olla ristiriidassa toisiinsa nähden (pitää tehdä poikkeus ja sitten poikkeuksen poikkeus ja niin edelleen). Alicen kohdalla näitä ongelmia ei tule, koska Alice muodostaa säännöt itse analysoituaan tietokannasta löytyvät kappaleet. Jos tietokantaan lisätään uusia kappaleita, voi Alice taas päivittyä samaa tahtia. Alice myös painottaa löytämänsä säännöt, eli jos tietokannassa on usein tapahtuvia liikkeitä/elementtejä musiikissa, Alice pyrkii käyttämään tällaisia liikkeitä/elementtejä myös luomassaan musiikissa. (Cope 2000, 103-107)

5.2 David Copen aikaansaannoksia

Liitteessä 2 listataan pieni joukko David Copen tekemiä musiikkikappaleita ja kirjoja, jotta lukija voi oman mielenkiintonsa mukaan tutkia aineistoa tarkemmin – esimerkiksi kuunnella Emin säveltämiä kappaleita YouTubesta. Mainittakoon vielä, että 2020 David Copelta on ilmestynyt tietokoneen kirjoittama novelli – eli tietokannassa on nyt musiikin sijaan tekstiä! ”Crushed Ice” -novellia pääsi lukemaan (viimeinen linkki liitteen 2 taulukossa) ainakin pari sivua maaliskuussa 2020, kun kävin Amazonin verkkokaupassa.

5.3 Muiden henkilöiden aikaansaannoksia

Liitteessä 3 listataan muita kuin David Copen omien ohjelmien säveltämiä kappaleita. Tyylit vaihtelevat klassisesta musiikista popmusiikkiin ja 'kokeellisempaan' musiikkiin. Halusin etsiä ja listata joitakin tietokoneen (ja ihmisen) säveltämiä kappaleita, jotta lukija saisi jonkinlaista kuvaa algoritmisen säveltämisen hyödyllisyydestä.

Omasta mielestäni (kaikki) kappaleet eivät ole yhtä hyviä kuin (parhaimpien) ihmisten tekemät kappaleet. Kun kuuntelee kappaleita, niin niistä löytyy kuitenkin todella paljon onnistuneita ratkaisuja muun muassa harmonian, rytmin ja melodian suhteen. Osa kappaleista on alkeellisen kuuloisia – vähemmän hienostuneita tekoälytekniikoita käyttävien ohjelmien tekemiä kappaleita. Mitä uudempia (tekoäly)tekniikoita hyödyntävien tietokoneiden säveltämiä kappaleita kuuntelee,

niin huomaa kuinka paljon (tekoäly)tekniikat ovat kehittyneet ja algoritmisen säveltäminen mennyt eteenpäin David Copen 2000-luvun alussa tekemästä (pioneer)työstä.

5.4 Algoritmisen säveltämisen hyödyt ja haitat?

Lukijalla on varmaan käynyt mielessä, mitä hyötyä ja haittoja algoritmisen säveltämisen käyttämisestä on. Näihin paneudutaan seuraavissa luvuissa. Luvussa 6 käsitellään algoritmisen säveltämisen hyötyjä. Luvussa 7 taas pohditaan mitä haittaa algoritmisen säveltämisen käyttämisestä voi olla.

6 ALGORITMISEN SÄVELTÄMISEN HYÖTYJÄ

Algoritmisella säveltämisellä on tärkeä merkitys musiikissa. Seuraavaksi listataan useita hyötyjä, joita algoritmisen säveltämisen käyttämisellä saavutetaan. Toivottavasti lukija vakuuttuu (edes jonkin verran) algoritmisen säveltämisen tärkeydestä ja hyödyllisyydestä näiden seuraavien esimerkkien avulla. Tätä lukuja seuraavassa luvussa, luvussa 7, kerrotaan algoritmisen säveltämisen haittoista.

6.1 Vertailu eri säveltäjien teosten kesken

David Cope toteaa, että hänen Emi ("Experiments in Musical Intelligence") ohjelmansa pystyy löytämään Mozartin sinfonioista paljon samankaltaisuuksia kun niitä verrataan Haydnin tekemiin sinfonioihin. Copen mukaan Mozartin sinfoniat 'kuulostavat' hyvin paljon Haydnin sinfonian kaltaisilta muun muassa muodon, tyylin ja orkestraation osalta. Emi-ohjelmasta löytyvän kaavamaisuuksia tunnistavan "pattern matching engine" -osan avulla Cope on löytänyt samankaltaisuuksia Mozartin sinfonioista, joissa Mozart on selvästi käyttänyt Haydnin sinfonoiden vastaavia osia hyödykseen. Cope mainitsee, että musiikin historiassa entisten mestarisäveltäjien tyylejä ja säveltämistekniikkaa on arvostettu ja heidän teoksiaan on käytetty esimerkkeinä, kun säveltämistä on opetettu uusille sukupolville. (Muscutt 2007)

6.2 Musiikkityyliin 'sisään' pääseminen helpottuu

Cope vahvistaa, että Emin tietokannat koostuvat tonaalisen ajan musiikista, joten lopputuloksena syntyvät tietokoneen avustamana tehdyt sävellykset ovat myös tonaalisia. Cope sanoo itse säveltävänsä enemmän atonaalisesti, joten hän pitää virkistävänä sitä, että on päässyt tonaalisen musiikin maailmaan 'sisälle' paremmin kun hän on päässyt kuulemaan ja tutkimaan Emin sävellyksiä. (Muscutt 2007)

6.3 Haarautumakohdassa mahdollisen jatkon valitseminen helpottuu

Cope miettii, että voisi säveltää omaa musiikkiaan ilman Emiä, mutta tämä olisi hänen mielestään resurssien haaskausta. Miksi? Koska Emin avulla hän pystyy hetkessä luomaan monta eri vaihtoehtoa mihin suuntaan lähteä seuraavaksi, jos hän on tullut musiikilliseen 'haarautumakohtaan'. Hän pystyy tutkimaan

Emin luomia vaihtoehtoja ja valitsemaan sitten ne, joita hän pitää parhaimpina kyseisessä tilanteessa. (Muscutt 2007)

6.4 Tietokone saa luotua lähtödataansa paremman teoksen

David Copelle esitettiin kysymys: "Onko Emi-tietokoneohjelmasi onnistunut säveltämään kappaaleen, jota kuuntelisit mieluummin kuin tietokantaan tallennettuja alkuperäisiä kappaaleita?". Cope vastaa, että on ollut hetkiä, jolloin jotain tuontapaista on lähes tapahtunut – eli Emin säveltämä kappaale olisi 'parempi' kuin tietokannasta löytyvä kappaale. Hän kuitenkin toteaa, että arvostaa alkuperäisen säveltäjän kappaletta paljon enemmän. Cope haluaa kuunnella ja arvostaa mitä tahansa kappaletta riippumatta siitä onko säveltäjänä ollut tietokone vai ihminen. Emi on säveltänyt kappaaleita yli 25 vuoden ajan ja osa kappaaleista on ollut sellaisia, että jotkut muusikot ovat halunneet esittää niitä konserttisaleissa, mutta heidän manageriensa kuultua säveltäjän nimen ovat joutuneet perumaan aikeensa. Managerit ovat näissä tapauksissa varoittaneet muusikoita ja kehottaneet heitä miettimään mitä tietokoneen tekemän sävellyksen esittäminen isoissa konserttisaleissa voisi tehdä heidän uralleen. Cope on pohtinut, miksi tietokoneen tekemiä sävellyksiä pidetään jotenkin vähemmän arvokkaampina kuin ihmisten tekemiä sävellyksiä. Selvää vastausta tähän hän ei ole saanut selville. Ehkä syy on siinä, että tietokone seuraa algoritmejaan ja tämä prosessi pysyy muuttumattomana sävellyksestä toiseen. (Muscutt 2007)

7 ALGORITMISEN SÄVELTÄMISEN HAITTOJA

Onko algoritmisesta säveltämisestä jotain haittaa? Voisiko se esimerkiksi vaikuttaa negatiivisella tavalla säveltäjään – esimerkiksi vähentää säveltäjän innoitusta, kun tietokone tarjoaa satoja (jopa tuhansittain) vaihtoehtoja johonkin ongelmaan? Seuraavaksi on listattu joitakin aiheeseen liittyviä ajatuksia algoritmista säveltämisestä työssään käyttäviltä säveltäjiltä.

7.1 Tietokoneen käyttö voi vähentää säveltäjän omaa ajatustyötä

Copen mukaan tietokonetta ei välttämättä tarvita algoritmisessa säveltämisessä. Hänen mielestään tietokoneet kuitenkin tarjoavat kätevän tavan toteuttaa algoritmit fyysisellä tasolla. Musiikin historian aikana paperille kirjoitetut algoritmit ovat toimineet ihan hyvin. Cope myös toteaa, että tietokone voi haitata säveltäjää, koska tietokoneella voi nopeasti kokeilla jotain ja säveltäjä voi valita liiankin helposti jonkin parametrin ja tämä voi Copen mukaan johtaa siihen, että säveltäjän ajattelu saattaa 'löystyä' ja sitä kautta lopputuloksen laatu laskea. Cope tarkoittaa, että paperia käytettäessä pitää olla heti 'hereillä' eli tietoinen siitä mitä tekee, koska homma pitää saada toimimaan heti ensi yrittämällä. (Muscutt 2007)

7.2 Tietokone voi luoda ison määrän kappaleita sekunneissa

Cope kertoo, että Emiä on käytetty jo yli 1000 tonaalisen musiikkiteoksen säveltämiseen. Beethovenin, Schubertin ja Bartókin käyttämien tyylien mukaiset sellokonsertot olisivat Copen mielestä mielenkiintoista kuultavaa ja Emi mahdollistaisi tällaisten kappaleiden luomisen. Cope on kuitenkin halunnut rajoittaa Emillä sävellettyjen kappaleiden määrää, koska Copen omien sanojen mukaan jossain vaiheessa 'liika on liikaa'.

Cope haluaa Emin kuuluvan historiallisten 'säveltäjien' joukkoon ja liian monen kappaleen säveltäminen himmentäisi tätä 'kuolemattomuutta'. Hän on poistanut Emin tietokantoja, mutta ei itse ohjelmaa. Eli periaatteessa olisi mahdollista saada Emi tekemään vaikka juuri edellä mainittu Beethovenin tyylinen sellokonsertto. Cope on kuitenkin päättänyt, että käyttää Emiä vain apuna omien kappaleidensa säveltämisprosessissa. (Muscutt 2007)

7.3 ”Tietokone vastaan ihminen”-vastakkain asettelu

Tulevaisuudessa voidaan olla tilanteessa, jossa Garry Kasparov oli 1990-luvulla – ihminen pelaamassa tietokonetta vastaan. Tuolloin Kasparov pelasi shakkia. Onko tulevaisuudessa luvassa vastaavanlainen taistelu ihmisen ja koneen välillä, mutta aihealueena ei ole shakin pelaaminen vaan musiikin säveltäminen? Näkevähkö ihmiset algoritmisen säveltämisen enemmän uhkana vai hyötynä?

Cope kertoo, että hän ei pidä Emi tietokoneohjelman säveltämiä kappaleita sellaisina, että ne kilpailisivat oikeiden ihmisten tekemien sävellysten kanssa. Hän painottaa, että ei pidä algoritmista säveltämistä ’ihminen vastaan tietokone’-kysymyksenä/asetteluna, vaan ennemminkin ’ihminen-kynän-ja-paperin-kanssa’ vastaan ’ihminen-ja-tietokone-apunaan’ -kysymyksenä. Tällä hän tarkoittaa sitä, että hän käyttää tietokonetta apuna säveltämisessä. Hän pitää hyödyllisenä sitä, että kun hän kirjoittaa ohjeitaan ja käskyjään koneelle, saa hän tulokset jopa 1000 kertaa nopeammin kuin jos hän olisi käyttänyt itse kynää ja paperia. Cope näkee siis algoritmisen säveltämisen mahdollistajana – ei uhkana. (Muscutt 2007)

Edwards toteaa, että yhä 2000-luvulla on paljon vastustusta liittyen algoritmiseen säveltämiseen – niin muusikoiden kuin tavallisten kansalaisten suunnalta. Tämä vastustus on Edwardsin mukaan samantyylistä mitä esiintyi aikanaan sarjallisuuden noustessa esille 1920-luvulla. Yksi syy voi myös olla se, että niin sanottu moderni musiikki ei miellytä ihmisiä. Edwards huomauttaa, että vastuksen syy on luultavammin siinä, että ihmiset luulevat tietokoneen säveltävän musiikin ihmissäveltäjän sijaan. Edwardsin mukaan suurimmassa osassa tapauksista algoritmiseen säveltämiseen liittyen ihmissäveltäjä on ollut suuressa roolissa kappaleen sävellystyössä – esimerkiksi ihminen on ohjelmoinut logiikan/säännöt sovelluksen käyttöön. (Edwards 2011)

8 VASTAUKSIA JOHDANTO-LUVUN KYSYMYKSIIN

Tutustuessani algoritmiseen säveltämiseen liittyvään kirjallisuuteen opinnäyte-työtä tehdessäni sain vastauksia moniin minua askarruttaneisiin algoritmiseen säveltämiseen liittyviin kysymyksiin. Listaan seuraavaksi Johdanto-luvussa mieleeni nousseet kysymykset ja vastaan löytämieni tietojen pohjalta:

- Kysymys 1: "Miten tietokone osaa säveltää? Osaako se säveltää?"
 - Vastaus: Tietokoneelle voidaan antaa iso määrä (laadukasta) dataa analysoitavaksi. Tästä lähtödatasta tietokone pyrkii löytämään yhdenmukaisuuksia ja eroavaisuuksia. Eli tietokone analysoi dataa ja muodostaa jonkinlaisen mallin/kuvan siitä miten lähtödata on koostunut ja mitä sääntöjä sen tekemiseen on käytetty – eli tietokoneen toivotaan löytävän tiettyyn musiikkityyliin liittyviä sääntönmukaisuuksia lähtödatan perusteella. Tietokonetta voidaan lähtödatan analysoinnin jälkeen pyytää säveltämään uusi lähtödatan 'tyylinen' kappale. Riippuen muun muassa tietokoneen käytämien algoritmien kehittyneisyydestä ja lähtödatan yhteneväisyydestä ja laadusta, on lopputuloksena syntyvä kappale enemmän tai vähemmän lähtödatan musiikkityylin tyyppinen.
- Kysymys 2: "Algoritminen/Tietokoneavusteinen säveltäminen... Mitä se on? Ja mitä se ei ole?"
 - Vastaus: Algoritmisen säveltämisen määritelmä voi vaihdella riippuen siitä keneltä kysyy. Olennaista ei ole tietokoneen käyttäminen, vaan se, että seurataan jotain algoritmia, että päästään haluttuun lopputulokseen. Kysymykseen siitä mitä se ei ole on vaikeampi vastata. Yksi asia mikä tulee mieleen, on se, että algoritmin pitää saada ratkaistua ongelma äärellisessä määrässä askelia. Jos kyseessä on ääretön määrä askelia, ei algoritmi pääse loppuun. Tällöin voisi väittää, että kyseessä ei ole algoritminen säveltäminen, koska sävellysprosessi ei koskaan loppuisi. Toisaalta algoritmi seuraa askeleitaan ja mahdollisesti luo jotain käyttäjälle näkyvää musiikillista materiaalia matkan varrella. Jää lukijan oman mielenkiinnon varaan pohtia mikä ei ole algoritmista säveltämistä. Itse uskon siihen, että kaikki säveltäminen on algoritmista – oli se sitten tiedostamatonta tai tiedostettua toimintaa.

- Kysymys 3: ”Pystyykö ihminen erottamaan (helposti) tietokoneen tekemän sävellyksen?”
 - Vastaus: Tietokoneen tekemä sävellys riippuu muun muassa siitä kuinka 1) hyviä algoritmeja on pystytty rakentamaan lähtödatan analysointiin, 2) kuinka laadukasta lähtödata on, 3) kuinka paljon lähtödataa on, 4) kuinka hyvin tietokoneen resursseja (kuten muisti) on käytetty hyödyksi ja niin edelleen. Jos tietokoneella rakennettu algoritmisen säveltämisen ohjelman on tarpeeksi hyvä, ei musiikin ammattilainenkaan välttämättä pysty sanomaan onko kappaleen säveltänyt tietokone vai ihminen. Eli riippuu käytettyjen algoritmien hyvydestä, kuinka hyvä lopputuloksena syntyvästä kappaleesta tulee. Esimerkiksi David Cope kertoo, että hänen tekemisään kuuntelijatesteissä eivät ihmiset, edes ammattilaiset, ole pystyneet erottamaan tietokoneen tekemää sävellystä ihmisen tekemästä.
- Kysymys 4: ”Jos tietokone osaa säveltää yhtä hienoa musiikkia kuin parhaimmat ihmissäveltäjät (kuten Mozart, Beethoven), tarvitaanko ihmistä/säveltäjää lopulta enää sävellysprosessissa ollenkaan?”
 - Vastaus: David Copelta on aikanaan kysytty sama kysymys ja hänen mielipiteensä oli, että ihmistä tarvitaan aina sävellysprosessissa mukana. Tietokone ei tule korvaamaan ihmistä vaan on ihmisen rinnalla – tukena auttamassa sävellysprosessissa. On hyvä myös pohtia sitä kysymystä, että kun monessa eri tekniikassa tarvitaan (ihmisen tekemää) lähtödataa analysoinnin tarpeisiin, niin miten tietokone voisi lähteä liikkeelle, jos sillä ei olisi lähtödataa ollenkaan? Jos tietokone osaisi tyhjästä taikoa uuden ihmisten mielestä laadukkaan kappaleen, niin omasta mielestäni siihen tarvitaan singulariteetin syntyminen – eli tietokoneesta tulisi kuin ihminen. Silloin olisimme luoneet jotain itsensä tiedostavaa ja silloin voisi todeta, että me olemme jumalia, jotka ovat luoneet tiedostavan olennon.
- Kysymys 5: ”Kuinka nopeasti tietokone säveltää kappaleen valmiiksi? Meneekö siihen sekunteja, tunteja, päiviä, viikkoa, kuukausia vai jopa vuosia?”

- Vastaus: Riippuu paljolti siitä minkälaisia algoritmeja on käytetty ja minkälaista kappaletta ollaan tekemässä. Lisäksi pitää miettiä pitääkö kappaleen olla laadultaan hyvä vai kelpaisiko esimerkiksi ihmisen mielestä keskinkertainen kappale. David Cope saa omilla ohjelmillaan hetkessä (satoja) tuhansia erilaisia tietokannan tyylin mukaisia kappaleita. Eli käytännössä ei kulu kuin pieni silmänräpäys kun tietokone tarjoaa uutta kappaletta ihmiselle ihmeteltäväksi.
- Kysymys 6: ”Kuinka paljon ihminen voi vaikuttaa tietokoneen sävellysprosessiin?”
 - Vastaus: Riippuu käytetystä ohjelmasta. Joissakin ohjelmissa pääsee vaikuttamaan pienempiinkin yksityiskohtiin, kun toisissa taas ihmisen vaikuttamismahdollisuudet kesken kappaleen tekoprosessin ovat vähäiset.
- Kysymys 7: ”Mitä ohjelmia on jo olemassa algoritmiseen säveltämiseen liittyen?”
 - Vastaus: Tutkiessani aihetta törmäsin siihen tosiasiaan, että nykyisin ohjelmia on paljon. Varsinkin tekoälytekniikoiden kehityttyä on uusia sovelluksia syntynyt 2000-luvun alkua enemmän. David Copenella on Emi- ja Alice-ohjelmat, Googlella taas Magenta-projekti. Jää lukijan oman mielenkiinnon varaan valita ohjelma(t) jo(t)ka kiinnostavat häntä ja lähteä sitten tutustumaan niihin tarkemmin.
- Kysymys 8: ”Onko tekoäly (englanniksi ’artificial intelligence’ eli AI) korvaamassa/syrjäyttämässä myös säveltäjän ammattikunnan jossain vaiheessa? Onhan esimerkiksi tavallisia rutiininomaisia töitä, kuten pellon auraus, jo teollisen vallankumouksen alusta siirretty ihmiseltä koneen avustuksella tehtäväksi.”
 - Vastaus: Viime vuosien tekoälyn ja siihen liittyvien tekniikkojen kehitys on kaventanut mielestäni välimatkaa tietokoneen ja ihmisen välillä sävellyksen tekemiseen liittyen. Tuntuu siltä, että tietokoneen opettaminen ajamaan esimerkiksi autoa on paljon helpompaa kuin laadukkaan sävellyksen tekeminen. En väitä tässä, etteikö tietokoneen ohjaama ’itseajava’ auto olisi helppo toteuttaa, vaan että sen tekemien on helpompaa kuin tietokoneen opettami-

nen säveltämään mestarillisesti. Tällä hetkellä tietokone toimii monin paikoin säveltäjän apuvälineenä ja tukee säveltäjää tämän sävellystyössä. Tekniikan ja tekoälyn kehittyessä en kuitenkaan pidä mahdottomana sitä, että jossain tulevaisuudessa tietokone osaa säveltää paremmin kuin ihminen on koskaan osannut. Tekniikan kehittymistä on kuitenkin vaikea arvioida ja jokainen saa muodostaa oman mielipiteensä algoritmisen säveltämisen tulevaisuudesta ja viekö esimerkiksi tietokone myös lopulta säveltäjien työn. Itse olen samaa mieltä kuin David Cope – tietokone on tukemassa ihmistä tämän sävellysprosessissa nyt ja tulevaisuudessa.

- Kysymys 9: ”Onko algoritmista säveltämisestä hyötyä säveltäjälle? Tulisiko tulevaisuuden säveltäjien tutustua tietokoneavusteiseen säveltämiseen, jotta työpaikka löytyisi AI-murroksen keskellä?”
 - Vastaus: Mielestäni algoritmisen säveltämisen periaatteiden tietämisestä on vain hyötyä säveltäjän kannalta. Olisi hienoa, jos jokainen säveltäjä löytäisi omaan käyttöönsä soveliaan (tekoäly)soveluksen säveltämisprosessin tueksi. En usko, että tällä hetkellä on tarpeen tietää algoritmista säveltämisestä – osaavat säveltäjät pystyvät kyllä säveltämään musiikkia ilman teknisiä apuvälineitä. Olen kuitenkin sitä mieltä, että jokaisen säveltäjän olisi hyvä tutustua siihen mitä algoritmisen säveltäminen on ja voisiko siitä olla apua omalla kohdalla.

9 POHDINTA

Oma tavoitteeni toteutui – sain tutustua itselleni aikaisemmin tuntemattomaan aihealueeseen eli algoritmiseen säveltämiseen. Mielestäni tämä opinnäytetyö tarjoaa sopivan ”pintaraapaisun” aihealueeseen ja saa lukijan toivottavasti pohtimaan omaa suhtautumistaan tietokoneavusteiseen/algoritmiseen säveltämiseen. David Copen mainitsema vastahankaisuus tietokoneen tekemiä sävellyksiä kohtaan näyttäisi nykyään hävinneen pikkuhiljaa – enää ei tarvitse (toivottavasti) pelätä ihmisten reaktioita, jos kertoo jonkin kappaleen syntyneen tietokoneen toimesta tai avustuksella.

Toivoisin, että tulevaisuudessa myös Suomessa opetettaisiin korkeakouluissa algoritmista säveltämistä. Kursseja soisi olevan ainakin valinnaisena, sillä tutustuttuani David Copen kirjaan ”The Algorithmic Composer” ja kuunneltuani joitakin tietokoneen säveltämiä kappaleita, kiinnostukseni algoritmiseen säveltämiseen kasvoi. Haluaisin käydä joitakin kursseja aiheeseen liittyen ja olisi hienoa, jos jossakin vaiheessa päästään siihen, että säveltäjien koulutuksessa on kursseja tarjota opiskelijoille algoritmiseen säveltämiseen liittyen.

LÄHTEET

Cope, D. 2000. The Algorithmic Composer. The computer music and digital audio series; Volume 16. Madison, Wisconsin: A-R Editions.

Edwards, M. 2011. Algorithmic Composition: Computational Thinking in Music. (artikkeli). Communications of the ACM, Vol. 54 No. 7. Luettu 24.2.2020
URL: <https://cacm.acm.org/magazines/2011/7/109891-algorithmic-composition/fulltext>

Järveläinen, H. 2000. Algorithmic Musical Composition – TML. (PDF). Luettu 24.2.2020
URL: <https://engineering.purdue.edu/ece477/Archive/2006/Fall/F06-Grp01/files/alco.pdf>

Kotilainen, S. 2017. Tekoälyn todellinen vallankumous. TiVi-lehti (eli TietoViikko-lehti) 5/2017, 18-25

Lorenzon, A. & Grönvall, J. & Brötjefors, K. & Olzon, P. & Andersson, O. & Wänerlöv, V. 2015. Algoritmisk composition av popmusik – utformad med markovkedjor, bayesiskt nätverk san strukturmodellering. Kandidatarbete inom Data- och Informationsteknik. Chalmers tekniska högskola. Göteborgs universitet.
URL: <http://publications.lib.chalmers.se/records/fulltext/219030/219030.pdf>

Muscutt, M. 2007. Composing with Algorithms: An Interview with David Cope. (artikkeli). Computer Music Journal, Vol. 31, Issue 3. Luettu 24.2.2020
URL: <https://www.jstor.org/stable/40072590?seq=1>
Käyttö vaatii käyttäjätunnukset (saa lukea ilmaiseksi 6 artikkelia kuukaudessa)

Ng, A. 2017. The State of Artificial Intelligence. YouTube-video. Katsottu 24.2.2020
URL: https://www.youtube.com/watch?v=NKpuX_yzdYs

Storås, N. 2017. Tekoälyn tutkijan testamentti. TiVi-lehti (eli TietoViikko-lehti) 5/2017, 26-29

LIITTEET

Liite 1. David Copen luento ”Computer synthesis of musical creativity”

https://www.youtube.com/watch?v=f_ZSR9E_8x0 Katsottu 25.2.2020

Tässä YouTube-videossa David Cope kertoo unelmastaan saada tietokanta aikaiseksi, missä on paljon Bachin koraaleja ja sitten rakentaa tietokantaa analysoiva ohjelma. Analysoinnin tuloksena hän voisi pyytää ohjelmaa säveltämään uuden Bach koraalin. Hän kertoo, että syötti yli 200 Bachin koraalia käsin tekemäänsä tietokantaan, koska siihen aikaan ei ollut tarvittavia työkaluja tämän työn automatisoimiseksi tietokoneella. (Cope vittaa luultavasti 1980-lukuun, koska silloin hän aloitti käsittääkseni algoritmiseen säveltämiseen liittyvän Emi-ohjelmansa tekemisen)

Cope hajoitti iskujen perusteella tahdin pieniksi osiksi analysointia varten. Sitten hän teki tietokoneella objekteja näistä pienistä osista (käsittelyn helpottamiseksi tietokoneella). Cope transponoi C-duuriin kaikki syöttämänsä Bachin koraalit ja alkoi kutsua iskujen kohdalta löytyviä sointuja 'lexiconeiksi' eli alkioiksi. Copen mukaan samoja sointuja pitäisi löytyä useammasta kohtaa eri Bachin koraaleissa – eli löytyisi säännönmukaisuuksia mikä sointu seuraa toistaan Bachin musiikkityylissä. Emi-ohjelma otti näitä alkioita eri koraaleista ja teki uuden koraalin 'leikkaa-liimaa'-menetelmällä. Sointujen analyysin jälkeen Cope lisäsi fraasin käsitteen ohjelmaansa, jotta uusi koraali olisi laulajille laulettavampi. Mitä pidemmälle mennään alkioden tutkimisessa, sitä tiukemmaksi säännöt menevät fraasin muodostamiselle.

Cope huomasi, että Bach teki 'virheitä' monissa kohtaa koraaleitaan, mutta kuitenkin harvoin. Oppilaita kielletään tekemästä tiettyjä asioita, jotta näitä virheitä ei tehtäisi 'koko ajan' – oppilaat tekevät enemmän virheitä kuin asiaan vihkiytynee opettajat/mestarit. Kuitenkin oleellista on se, kuinka usein 'virheitä' tehdään (todennäköisyys seuraavalle äänelle siirtymiselle). Cope jatkoi tekemällä oopperan kahdessa ja puolessa päivässä Emin avulla. Hän lisäsi myöhemmin lyriikat. Hän sai elämänsä parhaimmat arvostelut oopperasta. Hän ei kuitenkaan ollut tehnyt itse oopperaa – vai oliko? Cope huomasi sitten Bachin koraaleihin liittyen, että ohjelma ei käyttänyt yhtään taukoja! Cope lisäsi sitten ohjelmaan keinoja ottaa huomioon, missä kohtia taukoja pitäisi olla.

Copen mukaan tietokannassa pitää olla samanlaista 'tavaraa' eli joko duureja tai molleja, mutta ei molempia. Jos tietokantaan laittaa esimerkiksi kahdelle pianolle sävellettyjä kappaleita, niin ohjelma tarjoaa uusia kahdelle pianolle sävellettyjä kappaleita. Cope kertoo halunneensa ymmärtää paremmin rakastamaansa musiikkia, lisätäkseen tietyllä tyylillä sävellettyä musiikkia maailmaan ja herättääkseen keskustelua ihmisen ja koneen luovuudesta ja estetiikasta. Copen mukaan ihmiset eivät halua käyttää samoja tekniikoita mitä muut käyttävät eli halutaan tehdä asiat omalla tavallaan – myös säveltäjän työssä. Copen toteaa, että säveltäjät haluavat tulla tunnetuksi omilla taidoilla

Liite 2. David Copen tekemiä musiikkikappaleita ja kirjoja.

TAULUKKO 2. David Copen tekemää musiikkia ja kirjoittamia kirjoja.

Teos/Nimi	Tyyppi	Linkki
Computer synthesis of Musical Creativity	Luento	https://www.youtube.com/watch?v=f_ZSR9E_8x0 (LIITE 1)
Bernd M. Scherer with David Cope Interview	Haastattelu	https://www.youtube.com/watch?v=9puyNuMXkEw
Emily Howell fugue	Kappale	https://www.youtube.com/watch?v=jLR-_c_uCwI
David Cope Emmy Vivaldi	Kappale	https://www.youtube.com/watch?v=2kuY3BrmTfQ
David Cope Emmy Beethoven 2 beg	Kappale	https://www.youtube.com/watch?v=CgG1HipAayU
Bach style chorale Emmy David Cope	Kappale	https://www.youtube.com/watch?v=PczDLI92vlc
Albinoni Emmy Adagio76	Kappale	https://www.youtube.com/watch?v=JLKyDn5pl4o
Bach style Invention 15 Emmy Cope	Kappale	https://www.youtube.com/watch?v=-2suk4Jq9kM
Chopin style Nocturne Emmy	Kappale	https://www.youtube.com/watch?v=t6WeiyyAiYQ
Virtual Music: Computer Synthesis of Musical Style	Kirja	https://www.amazon.com/Virtual-Music-Computer-Synthesis-Musical/dp/0262532611/ref=sr_1_1?keywords=david+cope&qid=1582616566&sr=8-1
Hidden Structure: Music Analysis Using Computers	Kirja	https://www.amazon.com/Hidden-Structure-Analysis-Computers-Computer/dp/0895796406/ref=sr_1_5?keywords=david+cope+Computers+and+Musical+Style&qid=1582617678&sr=8-5
The Algorithmic Composer	Kirja	https://www.amazon.com/Algorithmic-Composer-Computer-Music-Digital/dp/0895794543/ref=sr_1_31?keywords=david+cope&qid=1582617137&sr=8-31
Experiments in Musical Intelligence	Kirja	https://www.amazon.com/Experiments-Musical-Intelligence-Computer-Digital/dp/0895793377/ref=sr_1_40?keywords=david+cope&qid=1582617384&sr=8-40

Computers and Musical Style	Kirja	https://www.amazon.com/Computers-Musical-Style-Computer-Digital/dp/0895792567/ref=sr_1_4?keywords=david+cope+Computers+and+Musical+Style&qid=1582617624&sr=8-4
Cope, D.: Computer Composed Music	CD	https://www.amazon.com/Cope-D-Computer-Composed-Music/dp/B0047E4MDG/ref=sr_1_30?keywords=david+cope&qid=1582617137&sr=8-30
Beethoven Symphony No. 10	CD	https://www.amazon.com/Beethoven-Symphony-David-Cope-Emmy/dp/B008PYPCN8/ref=sr_1_3?keywords=david+cope&qid=1582616566&sr=8-3
Crushed Ice: A Computer Generated Novel	Kirja	https://www.amazon.com/Crushed-Ice-Computer-Generated-Novel/dp/B0851MGWT9/ref=sr_1_4?keywords=david+cope&qid=1582616566&sr=8-4

Liite 3. Algoritmisen säveltämisen seurauksena syntyneitä kappaleita.

TAULUKKO 3. Algoritmisen säveltämisen seurauksena syntyneitä kappaleita.

Teos/nimi	Linkki
Recurrence – Music written by a Recurrent Neural Network	https://soundcloud.com/optometrist-prime/recurrence-music-written-by-a-recurrent-neural-network
DeepBach: harmonization in the style of Bach generated using deep learning	https://www.youtube.com/watch?v=QiBM7-5hA6o
A Common Humanity – AI Generated Music Composed by AIVA	https://www.youtube.com/watch?v=hmRzBVOF23M
a Eurovision song created by Artificial Intelligence: Blue Jeans and Bloody Tears	https://www.youtube.com/watch?v=4MKAf6YX_7M
Benoît Carré: Daddy's Car AI-composed music	https://www.youtube.com/watch?v=cTP0Sr_ehmY
Some music generated by char-rnn	https://soundcloud.com/o_oooo/some-music-generated-by-char-rnn

Liite 4. YouTube opetusvideoita.

TAULUKKO 4. YouTube opetusvideoita liittyen algoritmiseen säveltämiseen ja tekoälyn kehittämiseen.

Nimi	Selitys/kuvaus	Linkki
Can AI help me compose a hit song? Francois Pachet TEDx-ISTAlameda	Syväoppimisen hyödyntämistä eri musiikkityylien mukaisten kappaleiden tekemiseen. SONY:n "Flow Machines"-projektin projektipäällikkö kertoo kokemuksistaan.	https://www.youtube.com/watch?v=h6m_nUnwBbA
How computers are learning to be creative Blaise Agüera y Arcas	Videolla kerrotaan miten tekoäly on kehittynyt liittyen kuvien tunnistamiseen eli miten tietokone voisi tunnistaa esimerkiksi linnun kuvasta.	https://www.youtube.com/watch?v=uSUOdu_5MPc
Generating Songs With Neural Networks (Neural Composer)	Näytetään miten tietokone säveltää (peli)musiikkia neuroverkkojen avulla.	https://www.youtube.com/watch?v=UWxfnNXIVy8
AI evolves to compose 3 hours of jazz!	Tietokone säveltää jazz musiikkia neuroverkkoja hyödyntäen.	https://www.youtube.com/watch?v=nA3YOFUCn4U
EXAMINING: Is Artificial Intelligence The Future of Music?	Videolla (vuodelta 2019) käydään läpi erilaisia tekoälyyn liittyviä ohjelmistoja, jotka liittyvät algoritmiseen säveltämiseen.	https://www.youtube.com/watch?v=9z65GmphlKo