

Pauli Rosbacka

KULTTUURIALGORITMIT EVOLUUTIOALASKENNAN TEHOSTAMISESSA

Informaatioteknologian ja viestinnän tiedekunta
Kandidaattitutkielma
Toukokuu 2021

TIIVISTELMÄ

Pauli Rosbacka: Kulttuurialgoritmit evoluutiolaskennan tehostamisessa
Kandidaattitutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Toukokuu 2021

Kulttuurialgoritmit ovat evoluutioalgoritmien ja geneettisten algoritmien alalaji. Niiden avulla voidaan optimoida ratkaisuja eri alojen ongelmiin luonnonilmiöistä inspiroituneilla mekanismeilla, joita ovat esimerkiksi perinnöllisyys sekä geneettinen ja kulttuurillinen evoluutio. Kulttuurialgoritmit suoriutuvat empiirisen aineiston perusteella optimointiongelmista aiempia evoluutioalgoritmeja ja geneettisiä variantteja nopeammin, ja ovat varteenotettava vaihtoehto muille nykyaikaisille optimointimenetelmille.

Tutkielma keskittyy kulttuurialgoritmien toiminnan sekä niiden kannalta olennaisten käsitteiden ja konseptien tutkimiseen kirjallisia lähteitä koostamalla. Tutkielmassa esitellään myös kulttuurialgoritmien viimeaikaisia sovelluksia, ja tavoitteena on muodostaa tuore yleiskatsaus niiden käyttökohteista ja tulevaisuudennäkymistä.

Kulttuurialgoritmien paremmuutta evoluutioalgoritmien tai muiden optimointistrategioiden yli ei ole matemaattisesti todistettu. Esteenä todistamiselle on koko optimointikentän ongelmakeskeisyys, jossa tehokas ratkaisu voi lakata toimimasta, jos ongelman asetelua muutetaan. Kulttuurialgoritmien ja niiden hybridien suorituskyvyn tutkiminen ja vertailu muihin menetelmiin olisi aiheellista jatkossa.

Avainsanat: kulttuurialgoritmi, uskomusavaruus, metaheuristiikka, luonnonilmiöihin perustuva algoritmi, geneettinen algoritmi

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

Sisällysluettelo

1	Johdanto	1
2	Käsitteet	2
2.1	Metaheuristiikat	2
2.2	Darwinilainen evoluutio	3
2.3	Kulttuuri	4
3	Kulttuurialgoritmien määritelmä	5
4	Sovellusalueet	8
4.1	Tietotekniikka	9
4.2	Sähkötekniikka	10
4.3	Yhdyskuntatekniikka	10
5	Yhteenveto ja johtopäätökset	11
	Lähdeluettelo	13

1 Johdanto

Nykyinen moderni teknologiayhteiskunta on täynnä tekniikkaa ja infrastruktuuria, joiden toimintaa on mahdollista optimoida. Tämä tarkoittaa toiminnan muuttamista tehokkaammaksi esimerkiksi siten, että ajankäyttö tai energiankulutus saadaan minimiin, tai tuotanto maksimiin. Tällaisia reaaliaikailman kohteita ovat esimerkiksi tehtaiden tuotantolinjat, sähköiset komponentit ja logistiikka. Optimaalisia ratkaisuja ongelmiin voi olla vaikeaa intuitiivisesti keksiä. Eri ratkaisuja voidaan kuitenkin yrityksen ja erehdyksen kautta kokeilla, ja jos tarpeeksi resursseja ratkaisujen kokeilemiseen on, niin tällainen raa'an voiman haku voi olla tehokkain tapa löytää optimaalisin ratkaisu.

Tietokoneiden laskentateho on mahdollistanut raa'an voiman menetelmien monipuolisen käytön ongelmanratkaisussa. Yksi tällaisista laskentatehosta riippuvaisten menetelmien osa-alueista on Darwinin evoluutioteoriasta inspiroitunut evoluutiolaskenta (*evolutionary computing*), jonka kehitys sai alkunsa 50-luvulla, mutta jonka käyttöaste on noussut erityisesti 90-luvulla ja sen jälkeen. Evoluutiolaskennassa ratkaistavaan ongelmaan käytetään tietokoneen tehoa tuottamaan suuri määrä ratkaisuehdotuksia, joista optimaalisimmat valitaan tuottamaan itsensä kaltaisia perillisiä. Prosessia toistetaan, kunnes tarpeeksi tyydyttävä ratkaisu löydetään. Evoluutiolaskennan lisäksi puhutaan myös evoluutioalgoritmeista (*evolutionary algorithm*), jotka ajatellaan ensiksi mainitun osajoukkona. Ero on siinä, että evoluutiolaskentaan katsotaan kuuluvan myös eibiologiset evoluutorakenteet, toisin kuin evoluutioalgoritmeihin. (Hojjat & Nazmul, 2013)

Evoluutioalgoritmien pohjalta kehitettiin geneettiset algoritmit (*genetic algorithm*), jotka ovat edellä mainittujen yleisimpiä sovelluksia. Geneettisten algoritmien toimintaidea perustuu nimensä mukaisesti geeniteoriaan. Tämä näkyy algoritmeissa uusien operaatioiden muodossa, jotka mukailevat risteytymistä ja mutaatiota. Risteytymisoperaatio yhdistää eri vanhempien suosiollisina pidetyt ominaisuudet jälkeläiseen, ja mutaatio luo satunnaisesti uusia. (Kala et al, 2010) Tutkielmassa geneettisten algoritmien toimintaa ei tarkastella tämän enempää.

Kulttuurialgoritmit (*culture algorithm, cultural algorithm*) toimivat geneettisten algoritmien päivityksenä, vauhdittaen evoluutiota. Niiden tehokkuus perustuu populaatioon kohdistuvaan paineeseen, joka syntyy kulttuurin asettamista suorituskyyvyn rajoista, sekä yksilöistä erillisestä historiasta, johon tallennetaan yksilöiden suoritustiedot. Erityisesti funktio, joka tarkastaa, että onko yksilön suorituskyyky yleisesti hyväksytyllä tasolla, asettaa alarajan koko populaation suorituskyyvylle. (Reynolds, 1994) Yleisesti geneettisen algoritmin vahvuus piilee laskentatehossa, sillä tietokone voi ohittaa laskukykykoiden "elämänkaaren" kokonaan, keskittyen pelkästään periytymiseen, ja

käsitellä evoluutioprosessia paljon reaali maailmaa nopeammissa ajassa. Kulttuurialgoritmin asettamat rajoitteet nopeuttavat prosessia vielä enemmän.

Kulttuurialgoritmien paremmuutta evoluutioalgoritmien suhteen ei ole matemaattisesti todistettu, mutta empiirinen aineisto viittaa siihen, että evoluutioalgoritmin muuttaminen kulttuurialgoritmiksi parantaa suoritusnopeutta. Erityisesti ratkaisun optimirajan saavuttamiseen vaadittava aika lyhenee. Tämä on pääsyy kulttuurialgoritmin suosimiseen evoluutioalgoritmien ja geneettisten algoritmien yli. (Becerra & Coello, 2005)

Tutkielman tavoitteena on muodostaa yleiskuva kulttuurialgoritmien käytöstä 2020-luvulla, ja kartoittaa aiheita jatkotutkimukselle. Pohjatietoa ja artikkeleita on haettu Andor- ja IEEE-tietokannoista. Tutkielmassa esitellyissä sovelluksissa painotettiin niiden nykyaikaisuutta, kun taas vakiintuneissa määritelmässä ja pohjatiedossa nojattiin vanhempaan kirjallisuuteen. Kulttuurialgoritmien määritelmää tarkasteltaessa peruslähteinä toimivat Reynoldsin (1994) alkuperäinen konsepti, Becerran & Coellon (2005) kooste sekä Maherin (2021) uudempi kartoitus.

Kulttuurialgoritmeja käsittelevien tutkimusten ja sovellusten määrä on noussut 90-luvulta lähtien, ja 2020-luvulla niiden käyttöaste on terveellä tasolla. Optimointiongelmissa kulttuurialgoritmin soveltaminen on uskottava vaihtoehto. (Maheri et al, 2021) Mielenkiinnon kohteita kulttuurialgoritmien jatkotutkimuksessa ovat suorituskyky, monitavoitteisuus ja modulaarisuus.

Tutkielman rakenne on seuraavanlainen. Luvussa 2 esitellään aiheen kannalta olennainen käsitteistö, sekä kulttuurialgoritmeissa hyödynnettävät, luonnonilmiöihin perustuvat konseptit. Luvussa 3 käsitellään kulttuurialgoritmien toimintaa alkuperäisen kulttuurialgoritmin sekä myöhempien versioiden kautta, jotka eroavat hyvin vähän toisistaan. Luvussa 4 käydään läpi kulttuurialgoritmien sovellusalueita ja käyttöastetta, ja esitellään muutama esimerkkisovellus eri tekniikan aloilta. Lopuksi luvussa 5 on esitetty yhteenveto tutkielman sisällöstä sekä johtopäätöksiä tulevaisuudennäkymistä.

2 Käsitteet

2.1 Metaheuristiikat

Heuristiikka (*heuristics*) tarkoittaa ongelmanratkaisumenetelmää, jossa etsitään summittaista (*approximate*) ratkaisua, joka on tarpeeksi tyydyttävä, olematta kuitenkaan paras tai optimaalisin. Heuristiikat voivat useissa ongelmissa olla kustannustehokkain tapa löytää tyydyttävä ratkaisu, kuten NP-täydellisissä ongelmissa, jotka tarkoittavat lyhyesti laskennallisia ongelmia, joille ei ole olemassa polynomisessa ajassa toimivaa ratkaisua. Summittaiset algoritmit ovat suosittuja NP-täydellisten ongelmien ratkaisemiseen, sillä ne saavuttavat toimivan ratkaisun käytännöllisessä ajassa. (Rothlauf,

2011; Blum & Roli, 2003) Jos jokin menetelmä on heuristiikka, niin sen voidaan sanoa olevan *heuristinen*. Ei-algoritmisia heuristisia menetelmiä ovat esimerkiksi intuitiivinen päätöksenteko sekä valistuneet arvaukset. Heuristiset menetelmät toimivat myös hyvin yhdessä raan voiman hakujen kanssa, sillä heuristiikat voivat asettaa rajat hakualueelle alusta lähtien, tehden kaikkien objektien läpikäynnistä nopeampaa.

Heurististen menetelmien käyttö vaatii yleensä ennakkotietoa ratkaistavasta ongelmasta ja sen mahdollisista ratkaisuista. On myös tyypillistä, että heuristiikat, jotka ovat suunniteltu vain yhden ongelman ratkaisuun, ovat tehokkaita, kun taas useisiin ongelmiin tarkoitettut, yleisemmät heuristiikat ovat tehottomampia. Tehokkaimmat heuristiikat saattavat lakata toimimasta, jos ongelman asettelua vähänkin muutetaan. (Rothlauf, 2011) Heuristisille menetelmille on siis kysyntää sen verran kuin on olemassa ongelmia tai ongelmien variantteja, joihin eksaktin ratkaisun löytäminen on vaikeaa tai mahdotonta. Ongelmaksi nousee oikean heuristiikan löytäminen, joka voi pelkästään intuitiota käyttäen olla työlästä.

Hakuavaruus (*search space*) S on rajallinen joukko, joka sisältää kaikki mahdolliset ongelman P ratkaisut. Hakuavaruudella on osajoukko S^* , joka sisältää kaikki globaalisti optimaaliset ratkaisut, eli ne ratkaisut, jotka ovat jollain kriteereillä muita ratkaisuja parempia. Useimmiten globaalisti optimaalisin ratkaisu on laskennallisesti nopein. Ratkaisut koostuvat muuttujista, ja yksittäinen ratkaisuobjekti voi olla esimerkiksi luku, osajoukko tai graafirakenne. (Blum & Roli, 2003) Hakuavaruudella tarkoitetaan tästä lähin kaikkien mahdollisten heuristiikkojen joukkoa.

Metaheuristiikoille (*metaheuristics*) ei ole täysin yksiselitteistä määritelmää, mutta yleisesti ottaen niillä tarkoitetaan erilaisia algoritmeja tai strategioita, joilla voidaan etsiä tarpeeksi optimaalinen heuristinen hakumetodi. Optimointi tapahtuu tutkimalla hakuavaruutta iteratiivisesti, kunnes löydetään algoritmi, joka on lähimpänä optimaalista. Kuten heuristiikatkin, niin myös metaheuristiset algoritmitkin ovat summittaisia. (Blum & Roli, 2003) Lyhyesti metaheuristiikkoja voisi luonnehtia parhaiden etsintätapojen etsimiseksi.

Metaheuristiikat voidaan esittää yksinkertaisina korkean tason abstraktioina. Useat metaheuristiset menetelmät perustuvat luonnonilmiöihin, kuten esimerkiksi muurahaiskolonnaoptimointi sekä evoluutiolaskenta, jonka periaatteita käsitellään seuraavassa aliluvussa. Kumpikin edellä mainituista menetelmistä pohjautuu abstraktioihin, jotka jäljittelevät reaalimaailman luonnonilmiöitä. (Blum & Roli, 2003) Metaheuristiikat ja luonnonilmiöihin pohjautuvat algoritmit eivät kuitenkaan tarkoita samaa asiaa.

2.2 Darwinilainen evoluutio

Yksi luonnonilmiöihin perustuvien algoritmien alalaji ovat evoluutioalgoritmit, joiden inspiraationa toimii luonnossa esiintyvä evoluutio, eli lisääntymisen yhteydessä

tapahtuva seuraavan sukupolven rakentajiin kohdistuva muutos. Muutos jälkeläisissä tapahtuu joko mutaation tai rekombinaation kautta. Monilla yksilön ominaisuuksilla on mahdollisuus periytyä jälkeläisille, ja parhailla ominaisuuksilla on tapana yleistyä populaatiossa ajan myötä. Ominaisuuksien hyödyllisyys riippuu luonnonvalintana tunnetusta mekanismista. (Sivanandam & Deepa, 2008)

Luonnonvalinta tarkoittaa ympäristön muodostamia kriteerejä, joihin sopeutumalla yksilö todennäköisemmin selviytyy ja tuottaa omia jälkeläisiään. Hyvin ympäristöönsä sopeutuneiden eli *kelpoisten* yksilöiden ominaisuudet siis ”valitaan” jatkoon, kun taas heikommin sopeutuvien populaatiot joko supistuvat tai katoavat kokonaan. (Sivanandam & Deepa, 2008) Tyypillinen yksinkertaistettu esimerkki luonnonvalinnasta voisi olla paksun karvapeitteen yleistyminen eläinpopulaatiossa, kun ilmasto muuttuu pysyvästi kylmemmäksi. Eläimillä, joilla on ohut karvapeite, on suurempi todennäköisyys kuolla paleltumaan, jolloin ne eivät pääse lisääntymään. Eläimien, joilla on ohut karvapeite, määrä romahtaa, kun taas paksun karvapeitteen omaavat eläimet menestyvät.

On olennaista pitää mielessä, ettei evoluutio itsessään ole kehittymistä johonkin objektiivisesti parempaan suuntaan. Evoluutio on vain muutosta, ja kehittymisen vaikutelma johtuu ympäristöön sopeutumisesta luonnonvalinnan seurauksena. Ympäristö muodostaa lajeille vaatimukset, joihin parhaiten vastaavat lajin yksilöt selviävät ja pääsevät periyttämään näille vaatimuksille suosiollisia ominaisuuksiaan tuleville sukupolville. Tämä tulee olemaan tärkeää, kun siirrytään toteuttamaan näitä prosesseja laskennassa.

2.3 Kulttuuri

Richerson & Christiansen (2013) määrittelevät kulttuurin olevan opittua, opetettua tai imitoitua toimintaa, tapoja, ideoita ja normeja, jotka siirtyvät ihmiseltä toiselle. Kulttuuri on yleensä sidottu johonkin rajattuun ihmispopulaatioon, kuten yhteisöön tai valtioon. Kulttuurilla voi myös olla alakulttuureja, kuten ihmispopulaatiollakin voi olla alipopulaatioita, jotka eroavat toisistaan joillain osa-alueilla. Kulttuuria voisi luonnehtia tiedonlähteeksi, joka vaikuttaa kaikkiin kulttuurin alaisen populaation jäseniin, ja johon populaation yksilöt sijoittavat omaa tietoaan.

Kuten ihmispopulaatiotkin, myös kulttuuri muuttuu ajan ja eri sukupolvien myötä. Tätä kutsutaan kulttuurin evoluutioksi. Muutoksia voivat aiheuttaa mm. uudet keksinnöt ja ideat, jotka leviävät populaation sisällä. Myös tavat ja uskomukset voivat levitä populaation sisällä ja vaikuttaa yksilöiden toimintaan ja ajatusmaailmaan. Kokeneemmat yksilöt, tai tarkemmin ne, joilla on auktoriteetti populaatioon nähden jollain tietyllä osa-alueella, voivat vaikuttaa kulttuuriin enemmän kuin vähemmän kokeneet. (Richerson & Christiansen, 2013) Esimerkiksi vuodesta toiseen suurimman sadon populaationsa

keskuudessa saaneen maanviljelijän taitojen imitoimisesta on enemmän hyötyä selviytymisen kannalta kuin vähäisemmän sadon saaneen.

Kun kulttuurin evoluutio ja biologinen evoluutio tuodaan yhteen, voidaan muodostaa kaksinkertainen perinnöllisyysjärjestelmä. Kaksinkertaisessa perinnöllisyydessä tapahtuu kahdenlaista evoluutiota, jotka ovat mikroevoluutio (geneettinen perinnöllisyys), sekä makroevoluutio (kulttuurillinen perinnöllisyys), joista jälkimmäisessä yksilö perii kulttuurinsa kartuttamaa tietoa. (Becerra & Coello, 2005) Yksilöiden on mahdollista tulkita kulttuurin tietoa eri tavoin, riippuen heidän omista kokemuksistaan, johtaan erilaisiin lopputuloksiin yksilöstä riippuen. Eri tulkinnat voivat myös johtaa mutaatioihin verrannollisiin muutoksiin kulttuurissa.

Näiden määritelmien perusteella voidaan tiivistää, että populaatio käy geneettisen periytyksen lisäksi myös kulttuurin periytymistä, joilla on kummallakin vaikutusta toisiinsa. Poikkeusyksilöt voivat vaikuttaa kulttuuriin paljoltikin, ja kehittynyt kulttuuri vaikuttaa kaikkiin kulttuurin alaisiin ihmisiin. Tämä tulee olemaan olennaista, kun seuraavassa luvussa esitellään kulttuurialgoritmien ja uskomusavaruuden (*belief space*) toimintaa.

3 Kulttuurialgoritmien määritelmä

Kulttuurialgoritmin konseptin kehitti Reynolds (1994), jonka määritelmän mukaan kulttuurialgoritmi on laskennallinen malli, joka hyödyntää kaksinkertaista perinnöllisyyttä. Tarkemmin sanottuna tämä tarkoittaa sitä, että ratkaisujoukon lisäksi malliin kuuluu kulttuurin mallina toimiva uskomusavaruus (*belief space*), joista kumpikin käy läpi evoluutiota perinnöllisyyden kautta. Reynoldsin idean pohjalta kulttuurialgoritmeja on sittemmin kehitetty eteenpäin, ja hyödynnetty monilla eri aloilla, joista kerrotaan lisää luvussa 4. Kulttuurialgoritmien mallia voidaan soveltaa ongelmissa, joissa ratkaisujoukko voidaan kuvata populaationa.

Uskomusavaruus on kulttuurialgoritmien tapa kuvata kulttuuria. Siihen säilötään se yksilöiden kartuttama tieto, joka leviää käytöksen myötä seuraaville sukupolville. Kulttuurillista suorituspainetta voidaan kohdistaa populaatioon tyypillisesti tallentamalla sukupolven parhaat tulokset, ja vaikuttamalla näillä tuloksilla seuraavaan sukupolveen. Uskomusavaruuteen sisällytettävät tiedot jaetaan nykyään viiteen luokkaan, joista algoritmissa käytettävät luokat valitaan ongelman tarpeiden mukaan. (Maheri et al, 2021)

Uskomusavaruuden tietoluokkia ovat tilanteellinen (*situational*), normatiivinen (*normative*), topografinen (*topographic*), toimialueellinen (*domain*) ja historiallinen (*history, temporal*). Näistä vain tilanteellista tietoa hyödynnettiin alkuperäisen kulttuurialgoritmin uskomusavaruudessa. Siinä säilytetään sukupolven parhaiten suoriutuneiden yksilöiden tietoja. Normatiivinen tietoluokka kuvastaa kulttuurissa vallitsevia normeja ja käytäntöjä, ja tyypillisesti sisältää hyväksyttävän suoriutumisen

alarajan. Topografinen luokka lisää algoritmiin tilan mallinnuksen, ja kertoo tietoa ratkaisuyksiköistä sen perusteella, että miten ne tähän tilaan sijoittuvat. Toimialueellinen tietoluokka sisältää tietoa siitä, että millaiset ratkaisut ovat optimaalisina pidettyjä ongelman kannalta. Historiallista tietoa hyödynnetään, kun hakuavaruus muuttuu sukupolvien välillä. Siihen voidaan tallentaa tietoa hakuavaruuden ajanjaksoista ja parhaista ratkaisuksista. (Maheri et al, 2021) Taulukossa 1 on esitetty tietoluokat vielä yksinkertaisemmassa muodossa.

Taulukko 1. Uskomusavaruuden tietoluokat ja niiden lyhyet selitykset. (Maheri et al, 2021)

Tietoluokka	Selitys
Tilanteellinen tietämys	Sisältää sukupolven huomattavimmat tapahtumat, kuten parhaat ratkaisut.
Normatiivinen tietämys	Sisältää tietoa populaation hyväksyttävästä suoritustasosta, esimerkiksi alarajan suoritusajalle.
Topografinen tietämys	Kuvastaa ratkaisujoukkoa tilana, ja luokittelee ratkaisuyksiköitä niiden sijainnin mukaan.
Toimialueellinen tietämys	Sisältää tietoa ongelmasta, kuten optimaalisimpien ratkaisujen joukon.
Historiallinen tietämys	Tallentaa hakuavaruudessa tapahtuneet muutokset ja parhaat ratkaisut eri ajanjaksoilta.

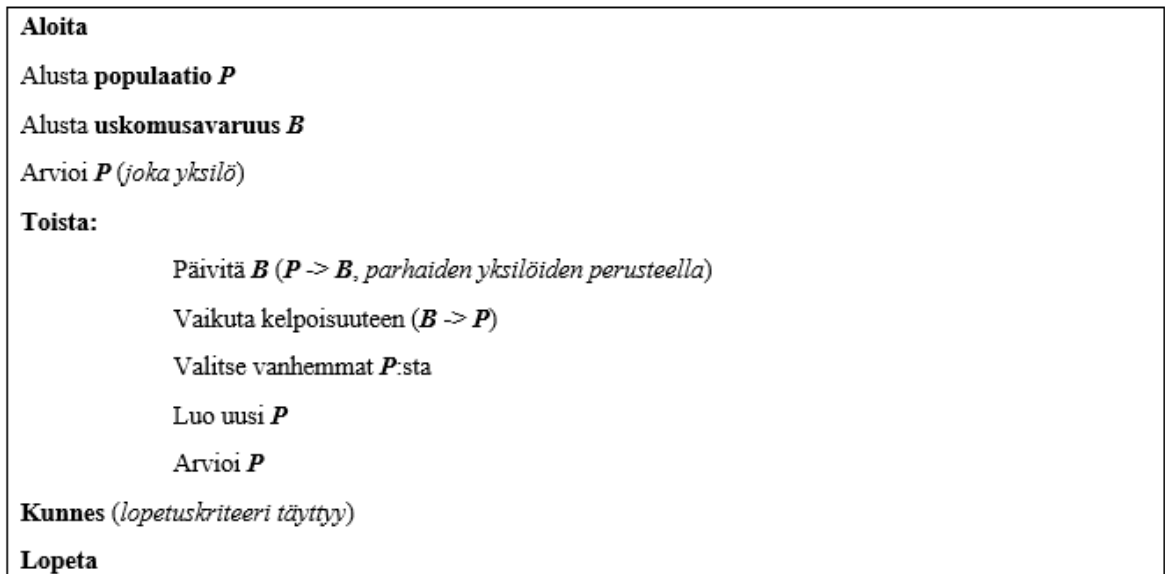
Luonnonvalinnan roolia toteuttaa kelpoisuusfunktio (*fitness function*), joka arvioi populaation jokaisen yksilön suorituskykyä joillain ongelmakohtaisilla kriteereillä, ja palauttaa kelpoisuusarvon. Kelpoisuusarvojen perusteella algoritmi tekee päätöksen, että mitkä yksilöt hyväksytään tuottamaan seuraava populaatio, ja että mitkä hylätään. Tyypillinen kelpoisuusfunktio laskee yksilön suorituskyvyn suhteessa koko populaation keskiarvoon, mutta funktion tarkempi toiminta riippuu ratkaistavasta ongelmasta. (Hojjat & Nazmul, 2013; Reynolds, 1994) Kelpoisuusfunktion suunnittelussa on aiheellista kiinnittää huomiota funktion suoritusajalle. Kelpoisuusfunktio ajetaan populaation jokaisen yksilön kohdalla erikseen, joten raskas funktio voi hidastaa algoritmin toimintaa tarpeettoman paljon.

Kulttuurialgoritmin vaiheet noudattelevat tyypillisten evoluutioalgoritmien suoritusta, mutta lisäksi algoritmiin ovat uskomusavaruuteen liittyvät toimenpiteet. Ohjelman suoritus alkaa alustavan populaation ja uskomusavaruuden luomisella. Populaatio täytetään satunnaisesti luoduilla ratkaisuyksiköillä. Alkupuolustuksen kaikki

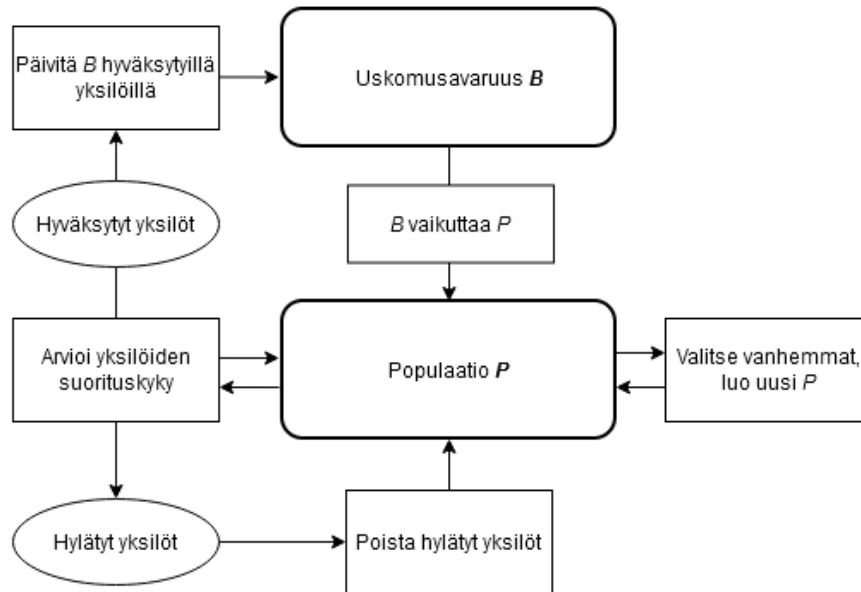
yksiköt arvioidaan kuntoisuusfunktiolla, ja hyväksytyt yksilöt vaikuttavat uskomusavaruuteen, esimerkiksi nostamalla ratkaisun suoritusajan alarajaa. (Becerra & Coello, 2005) Uskomusavaruuteen vaikuttavan funktion tarkka toiminta riippuu kuitenkin täysin ratkaistavasta ongelmasta.

Edellä mainittujen vaiheiden jälkeen tapahtuvat seuraavan yksikkösukupolven luomiseen vaadittavat toiminnot. Uskomusavaruuden päivittyneiden kriteerien mukaan valitaan vanhemmat, jotka toimivat seuraavan sukupolven pohjana. Parhaiten menestyneet yksilöt valitaan vanhemmiksi, samaan tapaan kuin uskomusavaruuteen vaikuttaessa. Hyväksymisprosessin tekevät funktiot voidaan jakaa kahteen tyyppiin, staattisiin ja dynaamisiin. Staattisessa hyväksynnässä populaatiosta tavallisesti valitaan jokin ennalta määritetty prosentti, kun taas dynaamisissa hyväksytyjen yksilöiden määrä vaihtelee sukupolvittain. Hylätyt yksilöt karsitaan algoritmin suorituksesta pysyvästi. (Reynolds, 1994; Becerra & Coello, 2005; Maheri, 2021) Menettely, jossa ratkaisuyksiköiden epätoivotuista ominaisuuksista pyritään pääsemään eroon ja toivottujen ominaisuuksien määrää pyritään lisäämään, vastaa läheisesti reaali maailman eugeniikkaa, niin positiivista kuin negatiivistakin.

Populaation seuraava sukupolvi luodaan täyttämällä uusi populaatio vanhempien ominaisuuksia perineillä ratkaisuyksiköillä, silti edelleen satunnaisesti. Populaatio arvioidaan jälleen, mutta koska kyseessä ei ole enää alkupopulaatio, niin kelpoisuusfunktio tarkistaa, että onko populaatiossa kriteerit täyttävä ratkaisu. Mikäli ratkaisuyksilö löytyy, niin algoritmin suoritus päätetään. Muussa tapauksessa algoritmin vaiheiden läpikäynti jatkuu uskomusavaruuteen vaikuttamisella, ja niin edelleen. Tällä tavoin muodostuu kulttuurialgoritmin silmukkarakenne. (Reynolds, 1994; Maheri, 2021) Erityisesti algoritmin keskeisen silmukan toimintaa selventämään kuvassa 1 on esitetty algoritmin toiminta pseudokoodina. Kuva 2 havainnollistaa kulttuurialgoritmin rakennetta. Kulttuurialgoritmien toimintamalli ei ole muuttunut vuosien 1994 ja 2021 välillä.



Kuva 1. Kulttuurialgoritmin toimintaa kuvaava pseudokoodi. (Reynolds, 1994; Becerra & Coello, 2005)

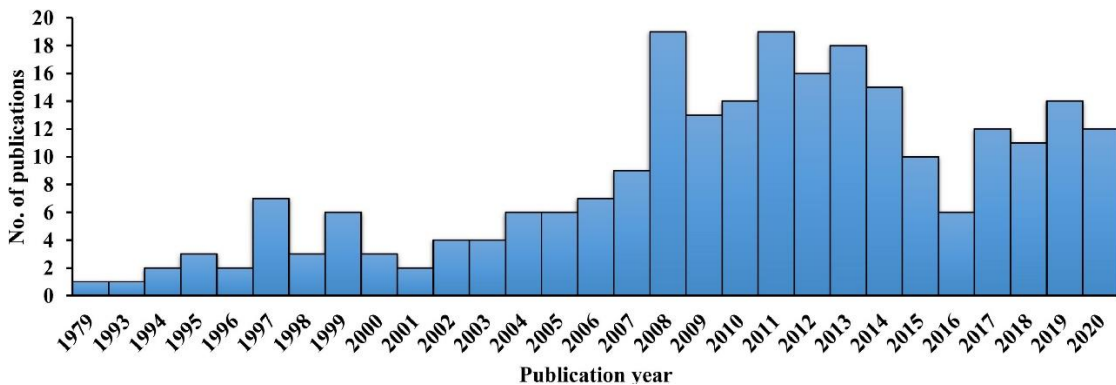


Kuva 2. Kulttuurialgoritmin rakenne. (Reynolds, 1994; Becerra & Coello, 2005; Maheri et al, 2021)

4 Sovellusalueet

Kulttuurialgoritmeihin kohdistunut kiinnostus on kasvanut sitten niiden keksimisestä vuonna 1994 nykypäivään saakka. Vuosien mittaan julkaisuissa on hyödynnetty kulttuurialgoritmeja monenlaisiin eri alojen ongelmiin, ja kehitetty uusia variantteja. Vuosien 1994 ja 2020 välillä yli puolet kulttuurialgoritmeihin liittyvistä julkaisuista olivat

tietotekniikan tutkimusalalta. Muita aloja olivat suurimmasta pienimpään sähkötekniikka, yhdyskuntatekniikka, konetekniikka ja muut alat yhtenä kategoriana. (Maheri et al, 2021) Kuvassa 3 on kuvattu vuosittaista kulttuurialgoritmeihin liittyvien julkaisujen määrää vuoteen 2020 saakka. Huippuvuodet ovat olleet 2008 ja 2011. Kulttuurialgoritmeihin kohdistunut kiinnostus tippui 2010-luvun puoliväliin mennessä, mutta sen jälkeen tutkimuksen määrä on noussut hieman ja jatkunut tasaisena.



Kuva 3. Kulttuurialgoritmeihin liittyvien julkaisujen määrä vuosittain (Maheri et al, 2021).

Seuraavissa aliluvuissa on esitelty uusia kulttuurialgoritmien esimerkkisovelluksia tietotekniikan, sähkötekniikan ja yhdyskuntatekniikan aloilta. Esimerkkien valinnassa pääkriteeri oli niiden uutuus, mutta myös sovellusalojen diversiteettiin kiinnitettiin huomiota.

4.1 Tietotekniikka

Abdolrazzagh-Nezhad et al (2020) kehittivät kulttuurialgoritmin optimoimaan monitavoitteisen attribuutinvähennysfunktion (*multi-objective attribute reduction, MOAR*) suoritusta. Attribuutinvähennyksellä poistetaan suurista tietojoukoista turhaksi määriteltyä dataa. Tätä tekniikkaa käytetään esimerkiksi tiedonlouhinnassa ja koneoppimisessa, joista kumpikin hyödyntää hahmontunnistusta. Vähennysfunktiolla on kaksi lopputavoitetta, jotka ovat optimaalisen pieni tietojoukko, tai mahdollisimman vähän poistettua tietoa. Monitavoitteinen attribuutinvähennys on NP-täydellinen ongelma.

Sovelluksessa käytetty kulttuurialgoritmi noudatteli pitkälti Reynoldsin alkuperäistä määritelmää. Algoritmissa käytetyt tietoluokat olivat tilanteellinen ja normatiivinen luokka. Sukupolvien välillä populaatiossa säilytettiin edellisen sukupolven yksilöt, jotka vaikuttivat uskomusavaruuteen, toisin kuin tyypillisessä kulttuurialgoritmissa. Heikosti suoriutuneet yksilöt korvattiin uusilla. (Abdolrazzagh-Nezhad et al, 2020) Menettelyä voisi tulevaisuudessa verrata klassiseen kulttuurialgoritmiin.

Toteutuksessa huomionarvoista olivat toistokokeet, joissa kulttuurialgoritmia verrattiin viiteen muuhun optimointialgoritmiin. Jokaista algoritmia testattiin 10 kertaa eri kokoisilla populaatioilla. Kulttuurialgoritmi suoriutui kaikissa toistoissa tehtävästä muita algoritmeja nopeammassa ajassa, mutta partikkeliparviälyn (*particle swarm optimization, PSO*) saavuttamat ratkaisut erosivat hieman vähemmän toisistaan. (Abdolrazzagh-Nezhad et al, 2020) Tämä on esimerkki siitä, että optimaalisenakin pidettyjen ratkaisujen joukossa varianssia voi esiintyä, kun mitään yksittäistä parasta ratkaisua ei ole.

4.2 Sähkötekniikka

Khorrani et al (2020) hyödynsivät kulttuurialgoritmia tehostaakseen hilan toimintaa. Hila on yksi tärkeimmistä optiikassa käytetyistä komponenteista, ja sen avulla valoa voidaan hajottaa ja jakaa eri aallonpituuksilla. Kulttuurialgoritmia käytettiin kehittämään hilan muotoilua niin, että valon heijastuvuusarvot ja geometriset seikat olivat optimaaliset. Optimoitavat rakenteet olivat monimutkaisia, mutta tämän sovelluksen perusteella kulttuurialgoritmi suoriutui tehtävästä asianmukaisesti.

Khorrani et al (2020) tiivistävät kulttuurialgoritmin olevan joustava ja yksinkertainen implementoida. Sitä testattiin myös geneettistä algoritmia ja partikkeliparviälyä vastaan, ja vertailussa se voitti kummatkin nopeudessa ja tulosten tarkkuudessa. Samoin kuin Abdolrazzagh-Nezhal et al (2020), niin tässäkin sovelluksessa käytetyt tietoluokat olivat normatiivinen ja tilanteellinen.

4.3 Yhdyskuntatekniikka

Kulttuurialgoritmeja on hyödynnetty myös yhdyskuntatekniikan alalla. Xue et al (2020) implementoivat uskomusavaruuden käkihakualgoritmin (*cuckoo search*) yhteyteen, ja käyttivät tätä hybridiä parantamaan ajoneuvojen reitinhakualgoritmia, jossa lisäongelmana oli matka-ajan ennustaminen muun liikenteen perusteella. Ajoneuvojen reitinhakuongelmassa (*vehicle routing problem, VRP*) ongelmana on vain nopeimman reitin valinta, mutta reaali maailmassa reitin kulkunopeuteen vaikuttavat valitun tien lisäksi myös muu liikenne, sääolosuhteet ja asiakkaiden tai tavaran määrä. Vaikutukset reaali maailmaan olisivat positiiviset, mikäli nämä muuttujat pystytään ennakoimaan ja ottamaan huomioon reitinhaussa.

Käkihauulla tarkoitetaan optimointialgoritmia, jossa hakuavaruus on kuvattu linnunpesien muodossa, joita on rajallinen määrä. Ratkaisuyksiköiden ajatellaan olevan pesissä olevien lintujen munia. Käki funktio sijoittaa pesiin uusia munia, jotka monipuolistavat pesien ratkaisutarjontaa. Parhaat pesät jatkavat seuraavalle sukupolvelle. Käkihaun ja kulttuurialgoritmin hybridissä parhaat pesät vaikuttavat uskomusavaruuteen, ja huonot pesät korvataan uskomusavaruuden pesillä. Tietoluokista hyödynnettiin tilanteellista ja normatiivista tietämystä. (Xue et al, 2020)

Tutkimuksessa kehitetyllä algoritmilla pystyttiin reaalitylanteessa ennustamaan ruuhkaisten teiden syntymistä Xiamenin kaupungissa. Tuloksista kävi myös ilmi, että kulttuurialgoritmin ja käkihaun hybridi menestyi paremmin kuin kummankaan pohja-algoritmin alkuperäinen versio. Se vaati vähemmän muokattavia parametreja, ja oli yleisesti ottaen nopeampi. (Xue et al, 2020) Nämä tulokset ovat lupaavia tulevaisuuden hybridiratkaisujen kannalta.

5 Yhteenveto ja johtopäätökset

Käsitellyn kirjallisuuden ja esimerkkisovellusten perusteella voidaan sanoa, että kulttuurialgoritmit ovat edelleen 2020-luvulla kilpailukykyinen vaihtoehto teknisten ratkaisujen optimoinnissa. Kulttuurialgoritmien etuja ovat implementoinnin helppous ja verrattain nopea suoritus aika. Heikkouksiksi voidaan laskea optimoinnin ongelmakeskeisyys, jonka takia globaalien kulttuurialgoritmin kehittäminen on mahdotonta.

Ongelmanratkaisussa käytettävät menetelmät valitaan ongelmanasettelun pohjalta, eikä todisteita löydy sille, että kulttuurialgoritmeja kannattaisi joka tilanteessa käyttää. Kun halutaan löytää jokin toimiva ratkaisu, niin useimmiten riittää vain tähän toimivaan ratkaisuun pääsy, eikä hakutuloksia vertailla käyttämättä jääneiden hakumenetödien vastaaviin. Vahvaa näyttöä kuitenkin löytyy sille, että aiemmat evoluutioalgoritmit ja geneettiset algoritmit ovat tehottomampia kuin kulttuurialgoritmit.

Tulevaisuudessa kulttuurialgoritmien suorituskyky tulee parantumaan ainakin niin kauan kuin tietokoneiden laskentateho kasvaa, tosin tällöin muutkin optimointimenetelmät tehostuvat. Nähtäväksi jää myös, että johtaako kasvava laskentateho optimointinopeuksien pienenemiseen mitättömälle tasolle, esimerkiksi kvanttietokoneiden kehittyessä ja yleistyessä. Jos hyviä tuloksia tulee joka optimointimenetelmällä saamaan vaikkapa vain joidenkin minuuttien laskennalla, niin silloin menetelmän tehokkuudella on vähemmän merkitystä kuin sen implementoinnin yksinkertaisuudella. Ratkaisutapojen ongelmakohtaisuuden vuoksi tulevaisuuden käytetyimmän menetelmän ennustaminen ei kuitenkaan ole mielekäästä. Historian perusteella on luultavaa, että kulttuurialgoritmi tulee pysymään vain yhtenä ratkaisutapana muiden joukossa.

Aiemmista tulevaisuudenvisionista huolimatta jatkossa tutkimustyötä olisi aiheellista kohdistaa kulttuurialgoritmien ja niiden varianttien tehokkuuden tarkempaan vertailemiseen suhteessa muihin optimointimenetelmiin. Myös uskomusavaruuden modulaarisuutta voitaisiin tarkastella muiden optimointimenetelmien kanssa. Jos todistusaineistoa löydetään sille, että uskomusavaruutta voidaan hyödyntää laajalti ja hyvin tuloksin muissakin algoritmeissa kuin ”puhtaissa” kulttuurialgoritmeissa, niin tästä seuraavat hybridialgoritmit voisivat potentiaalisesti korvata aiemmat strategiat.

Lähdeluettelo

Abdolrazzagh-Nezhad, M. et al. (2020). Enhanced cultural algorithm to solve multi-objective attribute reduction based on rough set theory. *Mathematics and Computers in Simulation*, 170(1), 332-350. <https://www-sciencedirect-com.libproxy.tuni.fi/science/article/pii/S0378475419303386>

Becerra, R., Coello, C. (2005). A Cultural Algorithm for Solving the Job Shop Scheduling Problem. Teoksessa Jin, Y. (toim.), *Knowledge Incorporation in Evolutionary Computation*. (s. 37-56) Springer.

Blum, C., Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys*, 268-308. <https://dl-acm-org.libproxy.tuni.fi/doi/epdf/10.1145/937503.937505>

Hojjat, A., Nazmul, S. (2013) *Computational intelligence: an introduction*. Wiley-Blackwell.

Kala, R. et al. (2010). *Towards Hybrid and Adaptive Computing: A Perspective*. Springer.

Maheri, A. et al. (2021). A comprehensive survey on cultural algorithms. *Swarm and evolutionary computation*, 62(2021). <https://www-sciencedirect-com.libproxy.tuni.fi/science/article/pii/S2210650221000079?via%3Dihub>

Reynolds, R. (1994). An Introduction to Cultural Algorithms. *3rd Annual Conference on Evolutionary Programming, Proceedings*. 131-139. https://www.researchgate.net/publication/201976967_An_Introduction_to_Cultural_Algorithms

Richerson, P., Christiansen, M. (2013). *Cultural Evolution: Society, Technology, Language, and Religion*. MIT Press.

Rothlauf, F. (2011). *Design of Modern Heuristics Principles and Application*. Springer.

Sivanandam, S.N, Deepa, S.N. (2008). *Introduction to Genetic Algorithms*. Springer.

Khorrami, Y., Fathi, D., Rumpf, R. (2020) Fast optimal design of optical components using the cultural algorithm. *Optics Express*. <https://www.osapublishing.org/oe/fulltext.cfm?uri=oe-28-11-15954&id=431742>

Xue, H. et al. (2020). Adaptive Cultural Algorithm-Based Cuckoo Search for Time-Dependent Vehicle Routing Problem with Stochastic Customers Using Adaptive Fractional Kalman Speed Prediction. *Mathematics problems in engineering*. <https://go-gale-com.libproxy.tuni.fi/ps/i.do?p=AONE&u=tampere&id=GALE|A642824709&v=2.1&it=r>