OHJELMATOTEUTUS GENEETTISESTÄ TAITEESTA

Jaakko Luttinen 62838F ja Janne Wallenius 60494V
16.12.2006

1. Johdanto

Yleisesti geneettisiä algoritmeja käytetään vaikeiden optimointiongelmien ratkaisemiseen. Niiden sovelluskenttää on mm. ohjelmointi, koneoppiminen, taloustiede, immuunijärjestelmä, ekologia ja genetiikka. Geneettisten algoritmien sisältämät tekniikat perintä, mutaatio, risteytys ja valinta ovat evoluutiobiologian inspiroimia. Käytännössä tarvitaan näiden metodien lisäksi alkupopulaatio, joka muodostetaan yleensä satunnaisesti, kuten myös tässä työssä.

Työssä on toteutettu Java-sovelluksena ohjelma, jolla voidaan generoida määrätyistä funktioelementeistä satunnaisesti rakentuvien satunnaispopulaatio, jonka jäseniä voidaan mutatoida ja risteyttää erilaisin operaattorein käyttäjän mieltymysten mukaisesti. Puhuaksemme taiteesta funktiopuiden palauttamat arvot esitetään graafisesti siten, että funktiopuu saa parametreinään pisteen koordinaatit ja palauttaa pisteelle värin intensiteetin.

2. Teoreettinen tarkastelu

Teoriassa geneettinen algoritmi etenee alkupopulaation määrityksen jälkeen siten, että populaation jäsenet evaluoidaan hyvyysfunktiolla (valinta). Tässä työssä hyvyysfunktion aseman korvaa ohjelman käyttäjän mieltymys. Seuraavaksi luodaan uusi populaatio geneettisten operaattoreiden avulla. Käytettäviä operaattoreita ovat erilaiset mutaatio- ja periyttämisoperaattorit, joissa käytetään vanhempina valittuja edellisen populaation jäseniä. Geneettisessä ohjelmoinnissa käytettyjä operaattoreita ei ole rakenteellisesti sidottu, mutta ideana on että periyttämisoperaattoreiden tuotteilla olisi vanhempiensa piirteitä. Mutaatio-operaattoreilla puolestaan olisi tarkoituksena tuoda populaatioon uusia piirteitä. Optimoinnin kannalta ajatellen mutatointi tarkoittaa geneettisille algoritmeille perinteisiin optimointimenetelmiin verrattuna sitä, että ei jumiuduta paikallisiin optimeihin.

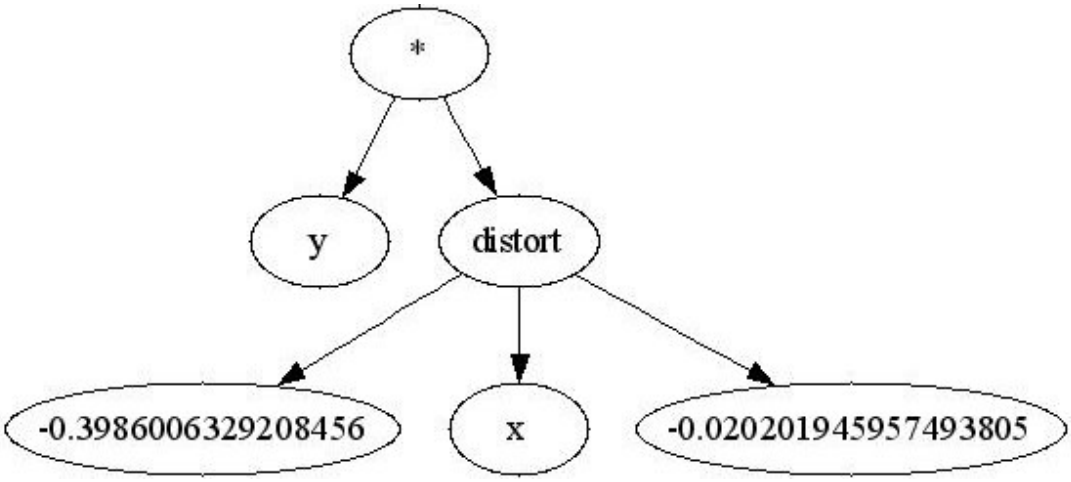
3. Käytännön toteutus

3.1 Funktiopuut

Ohjelmatyössämme populaation jäsenet ovat kuvan 1 kaltaisia funktiopuita. Puiden juurina toimivat koordinaatiston x ja y pisteet sekä satunnaisluvut. Näitä käytetään parametreinä ylempänä puissa oleviin funktioihin. Lopulta funktioketjut päättyvät, päähän, viimeiseen funktioon. Tämä palauttaa arvon, joka korreloi väri-intensiteettiä, jossain tietyssä (x,y)- koordinaattipisteessä.

Funktiopuissa on käytetty perinteisiä matemaattisia operaattoreita kuten itseisarvo, yhteen- ja vähennyslasku, kerto- ja jakolasku, sini, kosini, maksimi, minimi, x, y ja vakio. Lisäksi on käytetty kahta perinteisestä poikkeavaa funktiota. Toinen on distort, jolla on kolme lasta eli siitä lähtee kolme haaraa kohti juuria. Funktio ottaa kahden lapsensa arvot ja syöttää ne parametreinä kolmannen funktioon, ja palauttaa

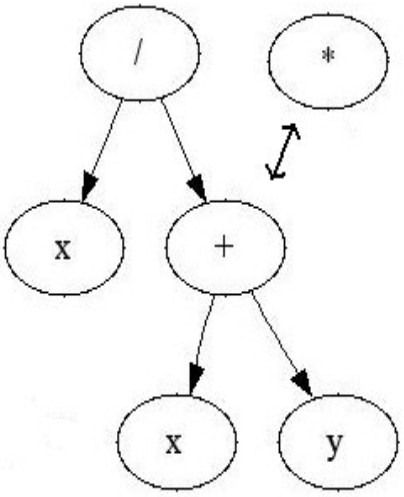
tämän arvon. Toisella, polar, voi olla vain yksi lapsi. Se muuttaa lapsensa saamat parametrit polaarikoordinaateiksi, säteeksi ja kulmaksi.



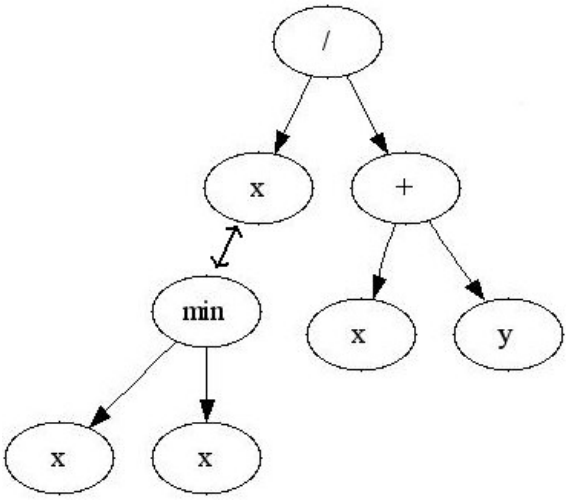
Kuva 1. Esimerkki funktiopuusta.

3.2 Mutaatiot

Ohjelmamme mutaatio-operaattori voi toimia kahdella tavalla, joko vaihtamalla yhden satunnaisen funktion toiseen satunnaiseen vastaavalapsiseen (Kuva 2) tai lisäämällä puun satunnaiseen juureen uuden pienen alipuun (Kuva 3). Operaattori arpoo kumpaa tietä mutaatio tapahtuu painotettuna funktion vaihdoksen todennäköisyyttä. Alipuun lisäyksellä näytti usein olevan radikaalimpi muutos alkuperäiseen kuvaan.



Kuva 2. Funktion vaihdos



Kuva 3. Alipuun lisäys

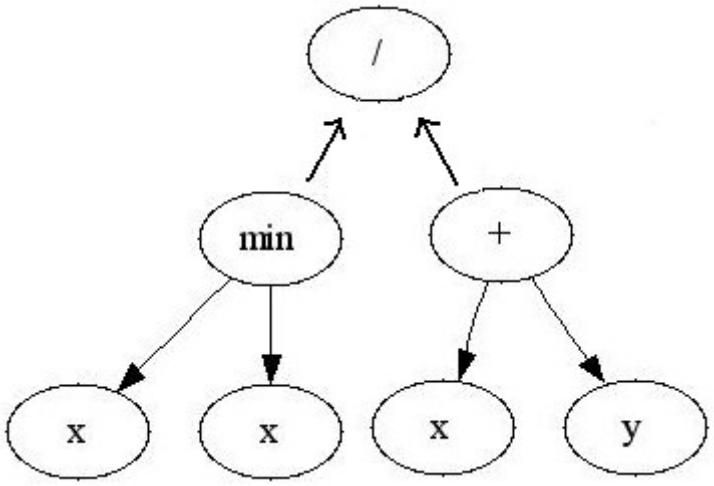
3.3 Risteytys

Käyttäjä voi valita tarpeen mukaan kolmesta erilaisesta risteytystyypistä. Tyypit ovat crossbreed simple, crossbreed seed ja crossbreed dominant. Crossbreed simple arpoo valitusta risteytettävien joukosta vanhemmat ja yhdistää ne satunnaisella funktiolla,

joka ottaa kaksi lasta, yhdeksi puuksi (Kuva 4). Crossbreed dominant arpoo vanhemmista toisen ikään kuin dominoivaksi. Dominoivan vanhemman pääfunktiolta erotetaan yksi lapsi eli haara, ja se korvataan toisen vanhemman puulla. Crossbreed seed –menetelmä on hieman monimutkaisempi. Se muodostaa vanhemmista ns. sukusolut, jotka ovat funktioita, joiden koko on puolet vanhempiinsa nähden. Tämä tiivistys tehdään siten, että funktiopuusta poistetaan osia. Jokainen sukusolun puun alkio on kuitenkin peräisin vanhemmasta. Kaksi sukusolua yhdistetään toisiinsa jonkin kaksilapsisen operaattorin avulla. Tässä menetelmässä lapsipuu on suunnilleen samankokoinen kuin vanhempien puut keskimäärin.

3.4 Käyttöliittymä

Käyttöliittymässä populaation kuvilla on omat pienet ikkunansa, ja niillä on omat komentonappinsa Z, N, S ja väri. Z avaa uuden ikkunan, jossa kuvan voi suurentaa (kuva täytyy kuitenkin aina piirtää uuteen kokoonsa uudelleen R-napilla). N luo uuden kuvan. S tallentaa kuvan kuvatiedostona tai puurakenteena, josta voi luoda havainnollisen kuvan erillisellä ohjelmalla (graphviz). Värit määräävät kuvan aktiivisuuden populaatiossa. Punainen on hylätty funktio ja vain niihin tapahtuu muutoksia operoidessa. Keltainen säilyttää funktion muuttumattomana operaattoreita käytettäessä. Vihreä on aktiivinen ja tällöin funktio sekä säilyy muuttumattomana että toimii vanhempana seuraavalle populaatiolle. Koko populaatioon vaikuttavat komentonapit ovat erillään kuvapopulaation ikkunoista. Näillä suoritetaan mutaatiot ja risteytykset. Jos kuvien piirtäminen on hidasta, tee ikkunasta pienempi, jolloin kuvistakin tulee pienempiä ja ne on nopeampi muodostaa.

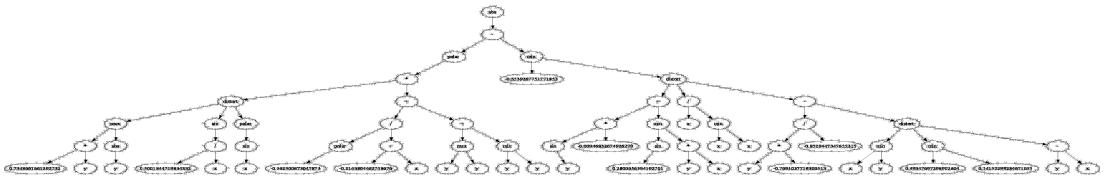


Kuva 4. Risteytystyyppi crossbreed simple

4. Tulokset

Käytännössä saadut puut olivat paljon suurempia kuin aiemmissa kuvissa esitetyt (Kuva 5). Lisäksi ne kasvoivat risteytysten yhteydessä. Itse kuvat ovat mustavalkoisia (Kuva 6). Eri risteytysoperaattorit tuntuivat onnistuneen sikäli hyvin,

että niiden vaikutus toisiinsa nähden on melko erilainen. Seed-risteytys ei synnyttänyt täysin vanhempiensa näköisiä jälkeläisiä, vaan pikemminkin yksilön, jossa oli kuitenkin havaittavissa tiettyjä piirteitä vanhemmistaan. Jos vanhempia on määritetty vain yksi, muodostuu ns. ”sukusoluja”. Riippuen vanhempien funktioiden rakenteista, tämä risteytys saattaa toimia hyvin tai huonosti. Dominant tuntui tekevän usein suoran yhdistyksen vanhemmista. Myös simple-risteytyksen jälkeläiset olivat hyvin vanhempiensa näköisiä.



Kuva 5. Puurakenne

Kuva 6. Hevimetal

5. Pohdintaa ja yhteenveto

Työ onnistui mielestämme hyvin. Käytetyt operaattorit tuntuivat lopulta tarkoituksenmukaisilta ja toimivilta, kuvistakin tuli ihan kauniita. Kehitettävää voisi värien kanssa eli värien lisääminen mukaan algoritmeihin jotenkin mielekkäästi toisi mukaan kokonaan uuden ulottuvuuden. Värien mukaanotto tarvitsisi myös tehokkaampia koneita ja menetelmiä, jotta kuvat piirtyvät nopeasti.

Geneettisessä ohjelmoinnissa käytetään yleensä suuria populaatioita. Tämä on hankala toteuttaa tällaisessa työssä, sillä ihminen ei jaksa katsoa keskittyneesti satoja tai tuhansia kuvia, eikä koneiden tehot riitä toimimaan näin suurilla populaatioilla tehokkaasti. Jos käytössä olisi kuitenkin tehokkaampia koneita ja/tai ohjelma olisi

toteutettu tehokkaammalla kielellä, voisi populaation koko kasvattaa hieman. Lisäksi puiden kokoa olisi mahdollista kasvattaa suuremmalla laskentateholla. Nyt noin 14 tasoa (eli n. 2^13 alkiota) kestää laskea jo selvästi havaittavan ajan. Tietenkin ikkunan koolla on merkitystä.

Ohjelmaan olisi hienoa toteuttaa ominaisuus, jolla funktioita voi tallentaa ja ladata uudelleen. Näin kerran luotu teos ei katoa, vaan sen voi avata ja jatkaa sen kehittämistä. Nyt kaikki funktiot tuhoutuvat, kun ohjelma suljetaan. Toki kuvia ja puurakenteita on voinut tallentaa, mutta ne eivät mahdollista funktion uudelleen avaamista.

Hienojen kuvien luominen ei ole helppoa. Yleensä alkupopulaatio on hyvin tylsän näköistä. Ahkeralla mutatoinnilla, risteyttämisellä ja tuurilla voi kuitenkin saada aikaan todella hienoja kuvia. Aikaa siihen saattaa kuitenkin kulua jonkin verran, ja käyttäjällä pitäisi olla jonkinlainen päämäärätietoinen näkemys siitä, minkä kaltaista kuvaa hän yrittää luoda. Mikään operaatio (mutaatio, generointi, risteytys) ei itsessään paranna populaatiota juurikaan, keskimäärin ehkä jopa huonontaa. Vauhdikkaalla naputtelulla ja kokeilemisella hienoja kuvia kuitenkin syntyy.

6. Lähdeluettelo

1. http://www.sal.tkk.fi/Opinnot/Mat-2.105/Geneettis.pdf, 15.12.06
2. http://steike.com/GeneticArt, 15.12.06

7. Liitteet

1. Koodit
2. Kuvia

Liite: Kuvia