

hyväksymispäivä arvosana

arvostelija

## **Evolutionääriset algoritmit**

Eetu Mäkelä

Helsinki 3.2.2003

HELSINGIN YLIOPISTO  
Tietojenkäsittelytieteen laitos

## Sisältö

<b>1</b>	<b>Evolutionääriset algoritmit</b>	<b>1</b>
<b>2</b>	<b>Geneettiset algoritmit</b>	<b>1</b>
<b>3</b>	<b>Evolutionääriset strategiat</b>	<b>2</b>
<b>4</b>	<b>Evolutionäärinen ohjelmointi</b>	<b>3</b>
	<b>Lähteet</b>	<b>3</b>

# 1 Evolutionääriset algoritmit

Evolutionäärinen laskenta ratkaisee optimointiongelmia luonnon evoluutiota mukailevien metodologioiden avulla. Evolutionäärisen laskennan vahvaa aluetta ovat monimutkaiset optimointiongelmat, joissa etsinnän suuntaa ohjaavia heuristiikkoja ei ole mahdollista muodostaa helposti, tai niiden avulla saadut tulokset osoittautuvat riittämättömiksi. Evolutionääriset algoritmit mukautuvat myös luonnostaan hyvin normaalisti ongelmallisiin muuttuviin ympäristöihin ja kesken laskennan vaihtuviin hyötyfunktioihin.

Evolutionääristen algoritmien perusidea on seuraava:

1. Algoritmi valitsee joukon mahdollisia ratkaisuja sattumanvaraisesti tai esitetojen pohjalta alkupopulaatioksi.
2. Ohjelma muodostaa nykyisestä populaatiosta uuden sukupolven yksilöiden risteytyksen ja mutaation avulla.
3. Ongelmaa varten rakennettu hyötyfunktio pisteyttää yksilöratkaisut
4. Algoritmi mallintaa luonnonvalintaa harventamalla ratkaisuja (useimmiten epäterministisesti) suhteessa niiden hyödyllisyyteen.
5. Algoritmi toistaa kohtia 2-4 kunnes se tuottaa tarpeeksi hyvän vastauksen tai käytettävissä oleva prosessointiaika loppuu.

Evolutionäärisen laskennan alueeseen kuuluvat algoritmit voidaan jakaa tarkemmin kolmeen päätyyppiin (jaon voidaan katsoa tapahtuvan sen tason mukaan, millä evoluutiota mallinnetaan): Geneettisiin algoritmeihin (geneettinen taso), evolutionäärisiin strategioihin (havaittujen ominaisuuksien taso) ja evolutionääriseen ohjelmointiin (lajitaso). joiden eroja käsittelemme seuraavaksi. Käsitteily pohjaa David B. Fogelin esittelyartikkeliin [Fog94].

## 2 Geneettiset algoritmit

Geneettiset algoritmit mallintavat evoluutiota geneettisellä tasolla. Yksilöt koodataan vektoreina, *kromosomeina*, joissa ratkaisuehdotelman jokaista parametria vastaa binäärikoodattu *geeni*.

Geneettinen algoritmi muodostaa uuden sukupolven yksilöihin kohdistuvien bittimutaatioiden sekä kahden yksilön välisten ristileikkausten (crossover) avulla. Tyypillisin käytetty ristileikkaus on yhden pisteen ristileikkaus, jossa ensimmäisen lisääntyvän geenin alku leikkauspisteeseen asti liitetään toisen lisääntyvän geenin loppuun ja toisinpäin. Muita leikkaustyypppejä ovat jatkuva ristileikkaus (uniform crossover), jossa lapsigeenin jokaisen bitin kohdalla luovuttava vanhempi arvotaan, sekä kahden pisteen ristileikkaus.

Yksilöiden ominaisuuksien koodaamista binäärisiksi geeneiksi on sanottu sekä geneettisten algoritmien vahvuudeksi että heikkoudeksi. Binäärikoodausta puolustavat vetoavat skeemateoriaan. Skeemateoria esittää, että jokaisen ratkaisun hyötyarvo antaa myös osviittaa kaikista muista ratkaisuista, joissa on edes joitakin samoja osia tai muotoja. Voidaan helposti osoittaa, että suurin määrä skeemoja käydään yhden ratkaisun arvioinnin kohdalla läpi kun koeteltavat ratkaisut koodataan binäärisiksi. Näin binäärikoodaus siis oletettavasti rinnakkaistaa etsintää maksimaalisesti.

Käytännön kokeet ovat kuitenkin asettaneet johtopäätöksen kyseenalaiseksi. Esimerkiksi reaalityöparametrien optimoinnissa usein tehokkaampi ratkaisu on saavutettu koodaamalla parametrit suoraan liukulukuina. Ylimääräinen koodaustaso tuo etsintäavaruuteen myös yhden ylimääräisen kerroksen monimutkaisuutta. Pienet muutokset geenissä välttämättä vastaa enää pieniä muutoksia parametrissa.

Yhden pisteen ristileikkausoperaation käyttöä lisääntymisvaiheessa on usein perusteltu *rakennuspalikkahypoteesilla* (*building block hypothesis*). Hypoteesin mukaan yhden pisteen ristileikkaus auttaa geneettistä algoritmia luomaan ja säilyttämään yhä pitempiä hyviksi havaittuja ”rakennuspalikoita”. Tästäkään ei kuitenkaan ole käytännön todisteita, vaan optimointikokeissa pikemminkin jatkuva ristileikkaus on osoittautunut paitsi yksinkertaisemmaksi, myös tehokkaammin hyviä ratkaisuja tuottavaksi lisääntymistavaksi. On myös huomautettu, että yhden pisteen ristileikkaus todennäköisemmin säilyttää yhdessä geenin keskiosan kuin geenin päissä olevat koodinpätkät.

### 3 Evolutionääriset strategiat

Evolutionääriset strategiat mallintavat evoluutiota geenitason sijasta yksilöiden havaittavien ominaisuuksien tasolla. Yksilöt määritellään useimmiten reaalityövektoreina, jokaisen luvun vastatessa jotakin optimoitavaa parametria.

Uusia sukupolvia tuotetaan pääasiassa mutaatiolla, kasvattamalla tai vähentämällä parametreja nollakeskisestä normaalijakaumasta poimituin satunnaismuutoksien. Normaalijakauman käyttö satunnaismuuttujan todennäköisyysjakaumana takaa sen, että parametreihin kohdistuu yhden lisääntymisjakson aikana todennäköisemmin pieniä muutoksia kuin suuria. Yksinkertaisimmassa  $(1+1)$  lisääntymisstrategiassa jälkeläisiä tuotetaan yksittäisistä yksilöistä.  $(\lambda+\delta)$  ja  $(\lambda,\delta)$  -strategioissa muodostetaan  $\lambda$  esi-isistä  $\delta$  jälkeläistä. Nämä kaksi strategiaa eroavat luonnonvalintaa mallintavalta vaiheeltaan.  $(\lambda+\delta)$  -strategia on *elitistinen*, parhaan ratkaisun varmasti säilyttävä, siinä selviytymisestä kamppailevat tasaväkisesti sekä vanhemmat että jälkeläiset.  $(\lambda,\delta)$  -strategiassa seuraavan sukupolven lisääntyjät valitaan taas ainoastaan  $\delta$  jälkeläisen joukosta.

Evoluutiostrategioissa tärkeitä laskentaa ohjaavia parametreja ovat mutaatiovaiheen muutoksiin käytettävien todennäköisyysjakaumien varianssit ja kovarianssit. Näiden parametrien optimaalisten arvojen arvioiminen etukäteen on usein hankalaa. Siksi pänekin otetaan usein mukaan muokattaviksi parametreiksi, yksilöiden ominaisuuksik-

si.

## 4 Evolutionäärinen ohjelmointi

Evolutionäärinen ohjelmointi toteuttaa evoluutiota lajitasolla. Populaatio muodostuu ohjelmista, jotka saavat ongelman syötteenään. Ohjelmat tulostavat oman ratkaisunsa ongelmaan, joka syötetään arviointifunktiolle.

Ohjelmien esitysmuodoksi metodologiassa on valittu tilakoneet. Lisääntyminen tapahtuu isäntäsupolven koneita mutatoimalla. Mutaatioita on viisi eri tyyppiä ja algoritmi soveltaa niitä useimmiten yhtä suurella todennäköisyydellä: Tilasiirtymän muuttaminen, tulostetun merkin vaihtaminen, tilan lisääminen, tilan poistaminen sekä alkutilan vaihtaminen.

Evolutionääristen ohjelmointialgoritmien on osoitettu toimivan hyvin vaativissakin tilanteissa. Esimerkiksi vangen dilemmaa mallintavassa testissä evolutionäärisellä ohjelmoinnilla tuotetut tilakoneet oppivat nopeasti arvioimaan populaation muiden koneiden toimintaa ja päätyivät hyvin nopeasti valitsemaan kaikille edullisen yhteistyötä painottavan käyttäytymismallin.

## Lähteet

- Fog94 Fogel, D. B., An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Networks*, 5,1(1994).