SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Slijedno pokretanje ECF-a

Igor Bespaljko

Voditelj: doc.dr.sc. Domagoj Jakobović

Zagreb, lipanj, 2010

**Sadržaj**

[1. Uvod 1](#_Toc159987575)

[2. Slijedno pokretanje ECF-a 2](#_Toc159987576)

[3. Zaključak 3](#_Toc159987577)

[4. Literatura 4](#_Toc159987578)

[5. Sažetak 5](#_Toc159987579)

# Uvod

ECF (Evolutionary Computing Framework) jest aplikacija, okruženje, za bilo koji tip evolucijskog računanja. Evolucijskim računanjem oponašamo prirodu u potrazi za rješenjima zadanih problema. Osnovni parametar evolucijskog računanja jest *populacija* nad kojom tražimo rješenje. Populaciju čini skup jedinki (u našem slučaju rješenja problema) nad kojima se provodi evolcijski algoritam. Algoritam iterativno primjenjuje operacije križanja i mutacije nad jedinkama populacije, iz generacije u generaciju, dok nije zadovoljen jedan od zadanih kriterija zaustavljanja algoritma, odnosno dok nije dobiveno zadovoljavajuće rješenje problema.

Korisnik, kako bi upotrijebio ECF za rješavanje problema, treba napisati, između ostalog, i konfiguracijsku datoteku u kojoj definira glavne parametre algoritma. Nakon pokretanja programa, čita se konfiguracijska datoteka i izvršava algoritam za zadane parametre. Prilikom izrade statistike o utjecaju promjene parametara (vjerojatnost mutacije, veličina populacije... itd) potrebno je više puta pokrenuti isti program i ručno mijenjati parametre, što uzima dosta vremena.

U ovome radu opisana je implementacija tzv. skupnog pokretanja ECF-a, što omogućuje automatsko izvršavanje algoritma proizvoljni broj puta uz promjenu nekih parametara.

# Slijedno pokretanje ECF-a

Osnovni parametar, za slijedno pokretanje, u korisničkoj konfiguracijskoj datoteci jest *batch.repeats*. Vrijednost tog parametra određuje koliko će se puta izvršiti algoritam. Npr. *<Entry* *key="batch.repeats">5</Entry>,* znači da će se algoritam izvršiti 5 puta. Vrijednosti određene batch prefiksom upisuju se pod *Registry* čvor u konfiguracijskoj XML datoteci. Ako je *batch.repeats* parametar izostavljen svi ostali parametri koji imaju batch prefiks nisu korišteni u algritmu. Osim broja ponavljanja algoritma korisnik može mijenjati vjerojatnost mutacije u samom izvršavanju programa. Npr. *<Entry key="batch.mutation.indprob">0.2 0.6 0.1</Entry>,* znači da se vrijednost mutacije mijenja od 0.2 do 0.6 u koracima od 0.1. Ako je u konfiguracijskoj datoteci naznačen gore navedeni broj ponavljanja algoritma te ako smo postavili prethodno spomenutu promjenu mutacije u koracima od 0.1, tada će se algoritam izvršiti svega 5 \* 5 = 25 puta. Još jedan parametar kojeg možemo promjeniti u toku izvršavanja algoritma jest veličina populacije. Korisnik mora navesti sve veličine populacije za koje želi provesti algoritam na sljedeći način: *<Entry key="batch.population.size">size1 size2... sizeN</Entry>*, gdje je *sizeN* n-ta veličina populacije za koju izvršavamo algoritam. Za sve gore navedene parametre, broj ponavljanja, promjena mutacije i populacije, algoritam bi se izvršavao 5\*5\*N puta. Primjetimo da se broj ponavljanja algoritma naglo povećava što je više batch parametara uključeno.

Svako izvršavanje algoritma se logira u zasebnu log datoteku, koja je označena sa *log\_xxx*, gdje „xxx“ znači iteraciju algoritma za svaku promjenu bilo kojeg parametra. Unutar log datoteke prvo su zabilježene vrijednosti parametara po kojima se trenutna iteracija razlikuje od drugih iteracija, zatim slijede rezultati algoritma za pojedine generacije. Na kraju svakog log file-a nalazi se statistika iteracije. U statistiku ulazi maksimalni, minimalni i prosječni fitness te standardna devijacija.

Ako korisnik želi ispis statistike slijednog pokretanja, dovoljno je postaviti u konfiguracijsku datoteku parametar *batch.stats* te se na kraju generira datoteka u kojoj se nalazi statistika svih iteracija kao i popis promjenjenih parametara pojedine iteracije. Parametar *batch.stats* se upisuje pod *Registry* čvor u konfiguracijskoj datoteci.

Glavna funkcija zadužena za slijedno pokretanje jest *runBatch()*, koja opetovano poziva funkcije za inicijalizaciju, *initialize(int argc, \*\*char)* i za pokretanje, *run().* Prije svakog poziva osvježavaju se oni parametri koji u konfiguracijskoj datoteci imaju prefiks *batch* te im se pridružuje nova vrijednost koju je također definirao sam korisnik.

Primjer konfiguracijske datoteke:

*<ECF>*

 *<Algorithm>*

 *<RouletteWheel>*

 *<Entry key="crxprob">0.5</Entry>*

 *<Entry key="selpressure">10</Entry>*

 *</RouletteWheel>*

 *</Algorithm>*

*<Genotype>*

 *<BitString>*

 *<Entry key="size">20</Entry>*

 *</BitString>*

 *</Genotype>*

 *<Registry>*

 *<Entry key="batch.repeats">2</Entry>*

 *<Entry key="batch.mutation.indprob">0.1 0.6 0.2</Entry>*

 *<Entry key="batch.population.size">50 100</Entry>*

 *<Entry key="batch.stats">1</Entry>*

 *<Entry key="randomizer.seed">0</Entry>*

 *<Entry key="population.size">30</Entry>*

 *<Entry key="mutation.indprob">0.3</Entry>*

 *<Entry key="term.maxgen">30</Entry>*

 *<Entry key="log.level">3</Entry>*

 *<Entry key="log.filename">log.txt</Entry>*

 *<Entry key="milestone.filename">out.txt</Entry>*

 *<Entry key="milestone.interval">0</Entry>*

 *</Registry>*

*</ECF>*

# Zaključak

 Slijednim pokretanjem ECF-a korisniku se omogućava jednostavno manipuliranje promjenjivim parametrima evolucijskog algoritma, bez potrebe zaustavljanja i ručnog mijenjanja parametara što bi zahtijevalo utrošak više vremena pri izvođenju pokusa i traženju samog rješenja zadanog problema. Korisnik jednostavno određuje parametre koji se mijenjaju tijekom izvođenja algoritma i njihove vrijednosti. Nakon završetka izvđenja programa, kao rezultat dobivaju se log datoteke iz kojih se, parametrom *batch.stats,* može lako izvući statistika što je veoma pogodno za različite analize.

# Literatura

 http://gp.zemris.fer.hr

# Sažetak

 ECF je programsko okruženje namjenjeno rješavanju problema pomoću evolucijskih algoritama. Osnovni parametar jest populacija jedinki nad kojom se provodi algoritam, odnosno traži rješenje primjenom različitih operatora algoritma (mutacija, križanje...itd.). Više o ECF-u pogledati u literaturi.

 Slijednim pokretanjem omogućeno je opetovano izvršavanje algoritma uz korisnički definiranu promjenu parametara. Korisnik jednostavno definira u konfiguracijskoj datoteci koliko iteracija algoritma želi te koje parametre želi mijenjati u svakoj iteraciji.

Dodavanjem parametra *batch.stats* korisnik dobiva ispis statistike za sve iteracije algoritma. U statistiku su uključene slijedeće vrijednosti: minimalni, maksimalni i prosječni fitness iteracije algoritma za zadane parametre ( veličina populacije, broj generacija i slično) te standardna devijacija i još neke vrijednosti vezane uz pojedinu iteraciju. Statistika može poslužiti za grafičko prikazivanje ovisnosti pojedinih parametara algoritma.