

Evolutionary Algorithms

Mick Bos en Youri de Vor

Uitleg Code

Call Tree

- main
 - population
 - individual
 - parentselection
 - fitness
 - newpopulation
 - crossover
 - mutationloop
 - bitmutation
 - answer
 - fitness

Totaal-Uitleg

De **main** loop wordt 100 keer gedraaid.

De loop begint met het aanmaken van een random populatie van gegeven grootte door middel van **population**. Population gebruikt **individual** die een random array genereert met een gegeven aantal waarden tussen een min en een max. In ons geval is dit dus een random samenstelling van 10 nullen en eenen.

Vervolgens wordt door middel van **parentselection** een aantal ouders gekozen. Parent selection maakt random pooltjes van gegeven grootte van alle individuals uit de populatie. Binnen een pool wordt alleen de individual met het beste antwoord geselecteerd. Dit wordt bepaald door middel van **fitness**.

Fitness berekend het absolute verschil tussen het antwoord en de target. Dit wordt gedaan door de absolute verschillen tussen de componenten van het antwoord en de target te sommeren.

Hierna wordt door middel van **newpopulation** een nieuwe populatie aangemaakt. Dit wordt gedaan door de lijst van geselecteerde ouders aan te vullen met kinderen tot dat de populatie grootte weer is bereikt. Kinderen worden gemaakt door middel van **crossover**. **Crossover** maakt een kind door per eigenschap met een kans van 50/50 te bepalen van welke ouder het kind de eigenschap overneemt.

Hierop volgt de **mutationloop**. In deze functie wordt gebruik gemaakt van **bitmutation**. Bitmutation gaat binnen een individu alle eigenschappen af en flipt deze wel of niet volgens een gegeven kans.

Aan het einde wordt het antwoord op de vraag gevonden door middel van **answer**. Answer berekent van ieder individual in de populatie de fitness en selecteert het antwoord van de individual met de beste fitness.

Vervolgens worden alle 100 antwoorden in een array gegooid en worden door middel van python statistics en numpy wat statistieken berekend.

Resultaten

In plaats van de antwoorden 100 keer te bekijken, bekijken we honderd keer de fitness van het beste antwoord. Onze fitness is immers letterlijk het absolute verschil tussen het antwoord en de target. Dit wordt berekend door het verschil te berekenen tussen de eerste stapel van het antwoord en die van de target en ditzelfde te doen voor de tweede stapel en dit te sommeren.

Een 1 betekent dus dat er een verschil van 1 punt was tussen het beste antwoord en de target. Een 0 betekent dat het beste antwoord overeen kwam met de target.

De runs zijn gedaan met default meta variabelen:

| Variabele | Waarde |
|-------------------|--------|
| Generaties | 100 |
| Mutatiekans | 0.0001 |
| Pool grootte | 4 |
| Populatie grootte | 1024 |

Hieronder zijn de default run en andere runs te zien. Andere runs worden gekenmerkt door hun afwijkende variabele in de bovenste rij.

| | Default | 400 Generaties | 0.001 Mutatiekans | 2048 Populatie grootte | 400 Generaties & 2048 Populatie grootte | 4096 Populatie |
|-----------------|-----------|-------------------|----------------------|------------------------------|---|-------------------|
| average | 0.13 | 0.14 | 0.13 | 0.05 | 0.04 | 0 |
| min | 0 | 0 | 0 | 0 | 0 | 0 |
| max | 1 | 2 | 2 | 1 | 1 | 0 |
| median | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| mode | 0 | 0 | 0 | 0 | 0 | 0 |
| st deviation | 0.3379977 | 0.402517 | 0.366667 | 0.219043 | 0.196946 | 0 |
| variance | 0.1130999 | 0.1603999 | 0.1331 | 0.0475 | 0.0384 | 0 |

average: 0 min: 0 max: 0 median: 0.0 mode: 0 st deviation: 0.0 variance: 0.0

Antwoord op de Vraag

Genotype

Het genotype wat wij gebruiken voor deze opdracht, zijn een selectie van 10 bits die aangeven of de kaart die hoort bij de bit, op stapel 0 of stapel 1 ligt. Hier hebben wij voor gekozen omdat dit voor ons de meest makkelijke implementatie leek. Ook is deze implementatie makkelijk te gebruiken bij de crossover en mutatie. Dit komt omdat je bij mutatie makkelijk de bit kan omdraaien en bij crossover is het een kwestie van kiezen van welke parent je de bit overneemt.

Statistieken en verwerking

Met de default setup was het eigenlijk al zo goed als goed genoeg. Met een gemiddelde fitness van 0.13 en een maximale fitness van 1 betekent dit dat het overgrote deel van de tijd exact het juiste antwoord gegeven wordt.

Door middel van een grotere populatie en een hoger aantal generaties kan het gemiddelde verder verlaagd worden naar 0.04. Dit betekent dat uit de 100 generaties slechts 4 keer een antwoord wordt gegeven dat niet perfect overeen komt met de target.

Op basis van deze bevindingen is er getest met een populatie van 4096. Deze populatiegrootte gaf 100 perfecte antwoorden. Alle statistieken waren dus nul.

Honourable mention

Poolsize en mutatiekans zijn ook kort getest. Poolsize had zo goed als geen invloed op de antwoorden en staat dus niet in de tabel. De mutatiekans vertienvoudigen verhoogde slechts de standaard deviatie en variance.