

Testaus

Paras tapa testata algoritmia on ajaa ohjelma ja katsoa kuinka se toimii. Käyttöliittymässä voi muokata eri asetuksia ja luoda sattumanvaraisia kartoja.

Ohjelmaa on myös testattu JUnitilla, mutta algoritmin random elementit estävät tietynlaiset testit, Koska kyseessä on approksimaatio algoritmi, myös testit approksimoivat tuloksen oikeellisuutta. Eksaktit osat, kuten pituuden laskeminen voidaan testata normaalisti, mutta kun koko algoritmia testaa, voi vain olettaa että ratkaisun pitäisi aina olla jonkin rajan alla. Joskus huonolla tuurilla vastaus on normaalia huonompi ja testit kaatuvat, vaikka algoritmi toimii oikein. Myös esimerkiksi mutatoitumista testataan vain tutkimalla onko perimä muuttunut. Usein mutaatiota tapahtuu, mutta ei aina.

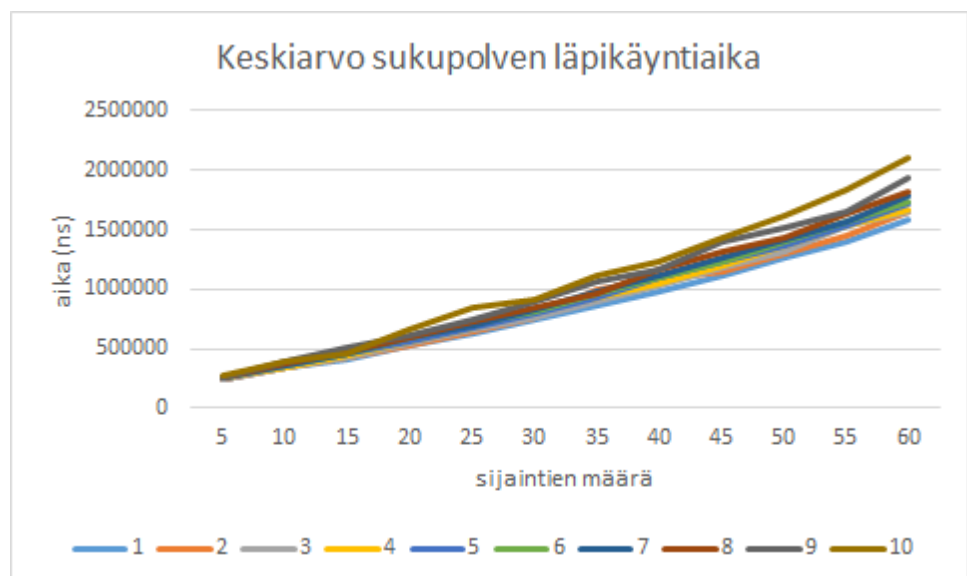
Ainoa tietorakenne joka algoritmia varten tarvittiin oli ArrayList. Normaalien JUnit testausten lisäksi tein testin joka tekee sattumanvaraisia komentoja ArrayListille. Tällä testillä voi varmistaa että mitkään normaalit syötteet eivät tuota odottamattomia virheitä. Testistä oli paljon hyötyä, koska se paljasti että koodissa on piileviä bugeja, tosin bugien löytäminen oli oma haasteensa.

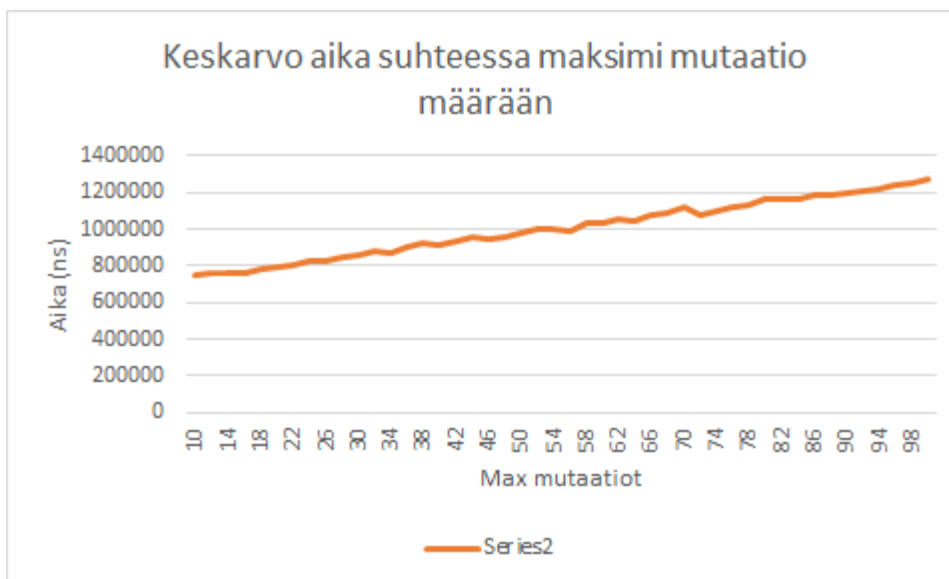
Suorituskyky

Ohjelma kykenee ratkaisemaan ongelman siedettävässä ajassa, jos sijaintien ja pakettien määrät pysyvät kymmenissä. Siedettävä aika on aika jonka pystyy odottamaan ilman että kyllästyy ja pysäyttää ohjelman.

Koska ohjelmassa on monta erilaista asiaa jotka vaikuttavat suorituskykyyn, testeissä syntyi suuri määrä dataa, vaikka arvot olivat suhteellisen pieniä. Suorituskyky testeissä laskin keskiarvo sukupolven luomis aikaa niin että autoja on käytössä 1-5, mutaatioita luotiin per DNA 10, 20 tai 30, paketteja per sijainti oli 1-5, ja sijainteja 5-60 5:n pykälissä. Ne arvot, jotka eivät vaikuta looppien määrään, pysyivät testeissä samoina. Näitä arvoja oli esimerkiksi fit- ja difdis, sekä pakettien painot ja autojen maksimi painot. Keskiarvo ajan laskin mittaamalla ajan tehdä 3000 sukupolvea ja jaoin aikavälin 3000:lla.

Testauksessa näkyy kuinka algoritmi käyttäytyy kovin lineaarisesti. Alla olevassa kaaviossa on esitetty sukupolven luontiaika suhteessa sijaintien määrään. Viivat esittävät kuinka monta pakettia jokaisella sijainnilla on. Autoja testissä oli 2 ja mutaatioita tehtiin maksimissaan 20.





Myös Mutaatioiden määrän muuttaminen kasvattaa aikaa lineaarisesti. Testissä käytettiin 30 sijaintia ja 5 pakettia per sijainti. Autoja oli 2.