Aalto-yliopisto
Perustieteiden
korkeakoulu

# Geneettiset algoritmit symmetrisen kauppamatkustajan ongelman ratkaisussa

*Jussi Hakosalo*

*04.12.2019*

Ohjaaja: M.Sc. *Juho Andelmin*
Valvoja: Prof. *Harri Ehtamo*



Systeemianalyysin
laboratorio

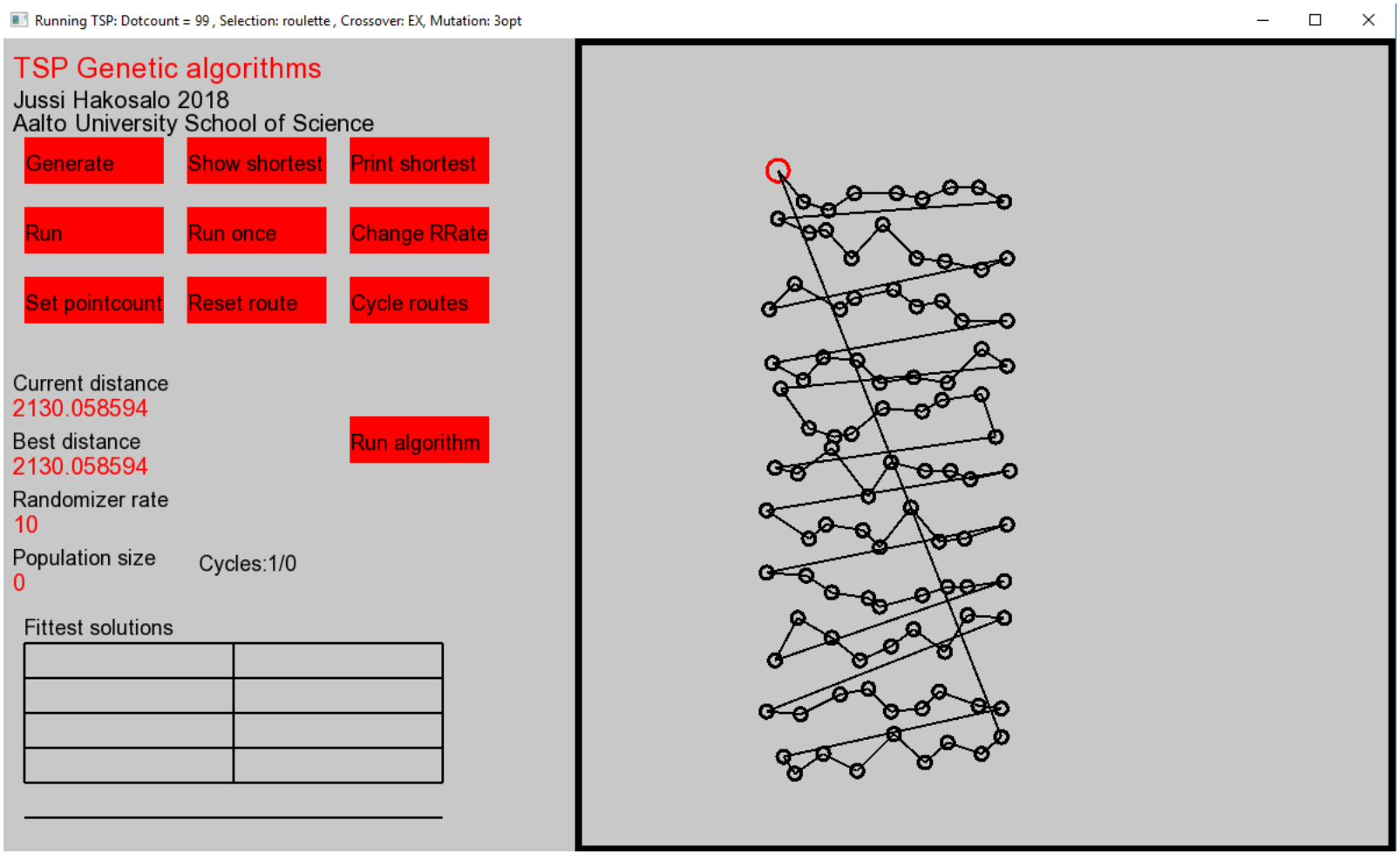
# Kertauksena: työn tavoitteet ja laajuus

- Tutkia kauppamatkustajan ongelman ratkaisemista geneettisillä algoritmeilla
- Tutkia muutamia yleisesti käytettyjä mutaatio-, valinta- ja risteytysmetodeja ja vertailla näiden suorituskykyä pienehköillä testi-instansseilla.
- Rakentaa graafinen simulaatiotyökalu, joka selkeyttää algoritmin toteuttamista ja toimintaa.
- Mahdollisesti myös tutkia valittujen populaatiokokojen ja lopetusehtojen vaikutusta algoritmin toimintaan.

# Simulaatiotyökalusta

- Huomattavasti aikaa vievin osa työtä oli simulaation rakentaminen (~2.5k LoC C++).
- Työkalu täytti tarkoituksensa mainiosti. Se mahdollisti instanssien ajamisen eri konfiguraatioilla tehokkaasti ja vaivattomasti.
- Simulaation lähdekoodi on GitHubissa julkisessa jaossa. Mahdollisuus käyttää myöhemmissä projekteissa, mikäli tulee tarpeeseen.

# Simulaatiotyökalusta



# Käytetyt instanssit

- Kokeissa käytetyt ongelmainstanssit valittiin noin 50-120 kaupunkia sisältäviksi.
- Kaupunkien lukumäärän kasvaessa ongelmat alkavat viemään huomattavasti enemmän laskenta-aikaa.
- Toteutetulla simulaatiotyökalulla laskenta-ajat olisivat venyneet kohtuuttoman pitkiksi isoimmilla graafeilla. Instanssit ovat sopivia tämän työn puitteissa.

# Käytetyt instanssit

| Instance name | Node count |
|---|---|
| berlin52 | 52 |
| bays29 | 29 |
| rat99 | 99 |
| kroA100 | 100 |
| lin105 | 105 |
| pr76 | 76 |
| pr124 | 124 |
| att48 | 48 |
| st70 | 70 |

# Simulaatioajoista

- Jokainen mainitusta yhdeksästä simulaatiosta ajettiin 3-4 eri konfiguraatiolla 5-10 kertaa ongelman koosta riippuen.
- Simulaatioissa käytettiin aloituspopulaation kokoa 70 ja muiden sukupolvien kokoa 40-70.
- Pysäytysehtoina käytettiin seuraavia:
  1. Nykyisen sukupolven reittien keskipituus on alle 2% pienempi kuin edellisen sukupolven vastaava.
  2. Nykyinen sukupolvi on järjestyksessä vähintään neljäs ja kahden edellisen sukupolven reitit ovat identtisiä.

# Simulaatiokonfiguraatiot

- Jokainen simulaatioajo koostuu kolmesta pääkomponentista: valintametodista, risteytysoperaattorista ja mutaatio-operaattorista.
- Valintametodit: Roulette selection (R), Elitism selection (Ru)
- Risteytysoperaattorit: Edge Recombination crossover (EX), Order crossover (OX)
- Mutaatio-operaattorit: Swap, Scramble, 2-opt (2opt), sekä 3-opt (3opt). On huomionarvoista, että Swap ja Scramble osoittautuivat testeissä erittäin huonoiksi, ja täten näitä ei käytetty lopullisissa simulaatioajoissa.

# Parhaat löydetyt ratkaisut

| Instance | Algorithm | Time/iteration | Avg Best | Best | Optimum | $\Delta optimal$ |
|---|---|---|---|---|---|---|
| berlin52 | Ru-EX-3opt | 26.06s | 7700 | 7544 | 7542 | 0.027% |
| att48 | R-EX-2opt | 44.34s | 33719 | 33523 | 33523 | optimality |
| bays29 | Ru-OX-2opt | 1.32s | 9116 | 9078 | 9074 | 0.044% |
| st70 | Ru-OX-2opt | 109.98s | 678 | 677 | 675 | 0.296% |
| kroA100 | Ru-EX-3opt | 422.196s | 21436 | 21285 | 21282 | 0.014% |
| lin105 | Ru-OX-2opt | 715.93s | 14506 | 14406 | 14375 | 0.216% |
| pr124 | Ru-OX-2opt | 1531.04s | 59527 | 59030 | 59030 | optimality |
| rat99 | Ru-EX-3opt | 495.62s | 1240 | 1223 | 1211 | 1.39% |
| pr76 | R-OX-2opt | 123.64s | 108825 | 108159 | 108159 | optimality |

# Tulokset

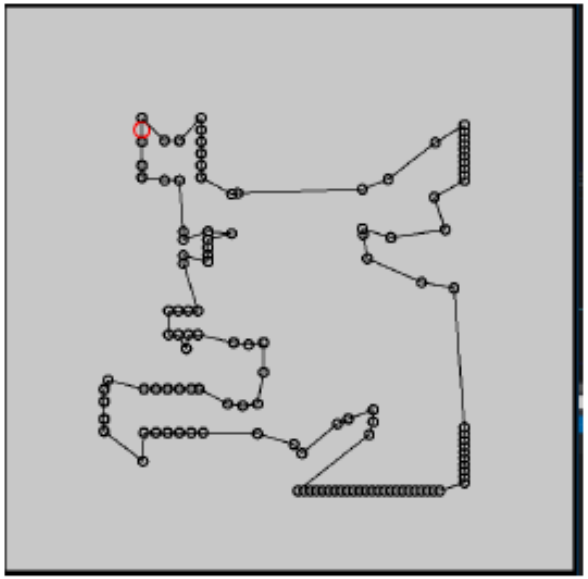


Figure 14: Pr124. Cost:59030. Roulette-OX-2opt

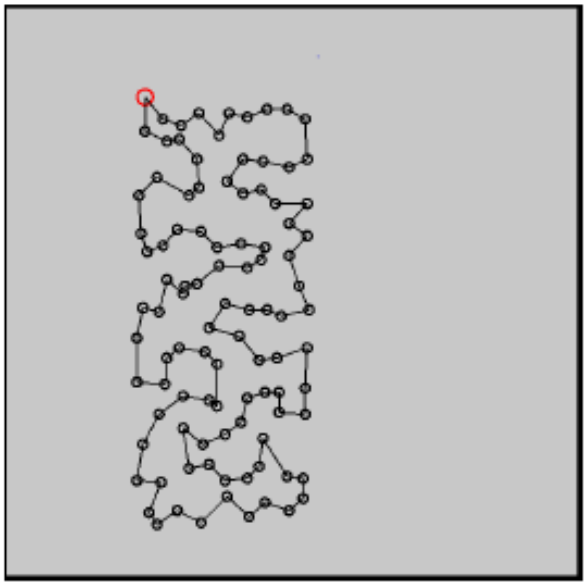


Figure 15: Rat99. Cost:1223. Roulette-EX-3opt

# Loppupäätelmät 1/2

- Tuloksista voimme nähdä, että algoritmi kykenee löytämään varsin hyvä ratkaisuja kohtuullisen nopeasti.
- Suurimmassa osassa instansseja päästiin prosentin osien päähän optimaalisesta ratkaisusta, muutamassa löydettiin optimi.
- Valtaosa laskenta-ajasta meni 2-opt ja 3-opt-mutaatioihin. Tuloksista nähdään, että 3-opt toimi nopeammin kuin 2-opt, vaikka ideaalitoteutuksessa 2-opt on $O(n^2)$ ja 3-opt $O(n^3)$. Tämä selittyy sillä, että 2-opt muokkaa reittiä huomattavasti useammin kuin 3-opt ennen konvergoitumista.

# Loppupäätelmät 2/2

- Tuloksista nähdään, että 2optia hyödyntävät algoritmit ovat 66% instansseissa toimivimpia. Tämä saattaa selittyä sillä, että 3opt tekee populaatioiden ratkaisuista liian homogeenisiä.
- Artikkelissaan[1)] Potvin toteaa, että OX ja EX-risteytysoperaattorit tuottavat parhaita tuloksia. Testien perusteella näiden kahden välillä ei ole merkittävää eroa. EX on kuitenkin marginaalisesti hitaampi kuin OX.
- Roulette selection tuotti parempia tuloksia Elitism selectioniin verrattuna 78% instansseissa. Elitismivalinta vaikuttaa tekevän populaatioista homogeenisempia, mikä eliminoi heikompien yksilöiden potentiaalisia hyötyjä. Tämä on todennäköinen syy havaittuihin paremmuuseroihin.

1) Potvin, J. Genetic algorithms for the traveling salesman problem. Annals of Operations Research 63(1996) 339-370.

# Puutteita ja parannuksia

- Ylimääräisen mutaatiokerroksen lisääminen algoritmin loppuun voisi lisätä populaation monipuolisuutta. Toinen idea on rulettivalinnan painotusten muuttaminen.
- 2opt ja 3opt operaattorien profilointi ja tarkempi tehokkuusvertailu.
- 2opt ja 3opt operaattorit hyödyntävät rinnakkaislaskentaa, mutta koodia ei ole muuten optimoitu. Tätä voi parantaa optimoimalla tietorakenteiden käyttöä sekä käyttäen hyväksi GPU-laskentaa.
- Työn puitteissa ei ollut aikaa suorittaa laajempia testiajoja usealle instanssille. Laajempi testaus on prioriteettina jatkokehitykselle.

# Seuraavat askeleet

- Lisää testausta laajalla joukolla instansseja: tehdyt kokeet olivat suppeita ja määrällisesti vähäisiä.
- Ohjelmaa toteutusta tulisi tehostaa siten, että isompia instansseja pystytään ratkaisemaan järkevässä ajassa.
- Keld Helsgaunin muokattua Lin-Kernighan-heuristiikkaa pidetään parhaimpana heuristisena algoritmina TSP-ongelmien ratkaisemisessa. Tämän implementointi tehostaisi algoritmin toimintaa huomattavasti.

# Kiitos!

https://github.com/hakosaj/KandiTSP