Kysyntäohjautuvan joukkoliikenteen matemaattisia malleja ja algoritmeja

Lauri Häme

Aalto-yliopiston perustieteiden korkeakoulu

27. huhtikuuta 2013

Johdanto

- Kysyntäohjautuva joukkoliikenne = bussi- ja taksipalvelujen välimuoto, joka perustuu ajoneuvojen joustavaan reititykseen
- Väitöskirjan tärkeimpiä tuloksia ovat älykkäät reitinlaskentamenetelmät
- Menetelmiä voidaan joukkoliikenteen lisäksi hyödyntää esim. rahti- ja lentoliikenteessä, lähetti- ja ruoankuljetuspalveluissa sekä sotilaslogistiikassa

Kysyntä & tarjonta

- Ajoneuvojen reitinlaskenta, ohjauslogiikka
  - Esim. Kutsuplus
- Matkustajien matkansuunnitteluongelma
  - Esim. Reittiopas
- Taloudellinen tasapaino

Kauppamatkustajan ongelma

- Tunnetuin reitinlaskentaongelma on ns. kauppamatkustajan ongelma (Traveling Salesman Problem, TSP)
  - Joukko maantieteellisiä pisteitä, joiden väliset etäisyydet tunnetaan
  - Tavoitteena on löytää lyhin reitti joka kulkee kaikkien pisteiden kautta
  - Laskennallisesti haastava ongelma

Kauppamatkustajan ongelma, esimerkki

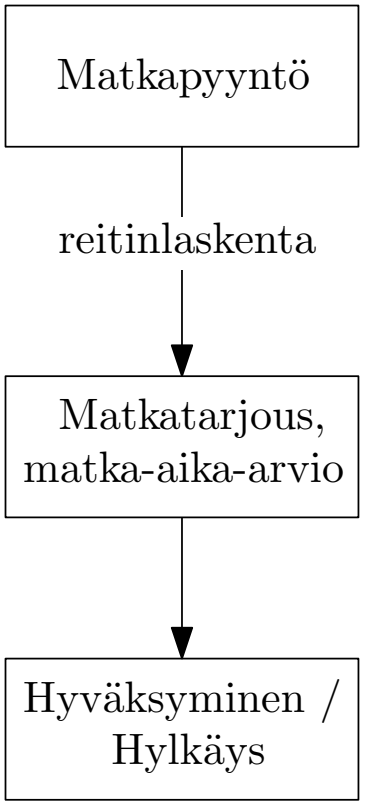
Kauppamatkustajan ongelma, esimerkki

Reitinlaskenta kuljetuspalveluissa

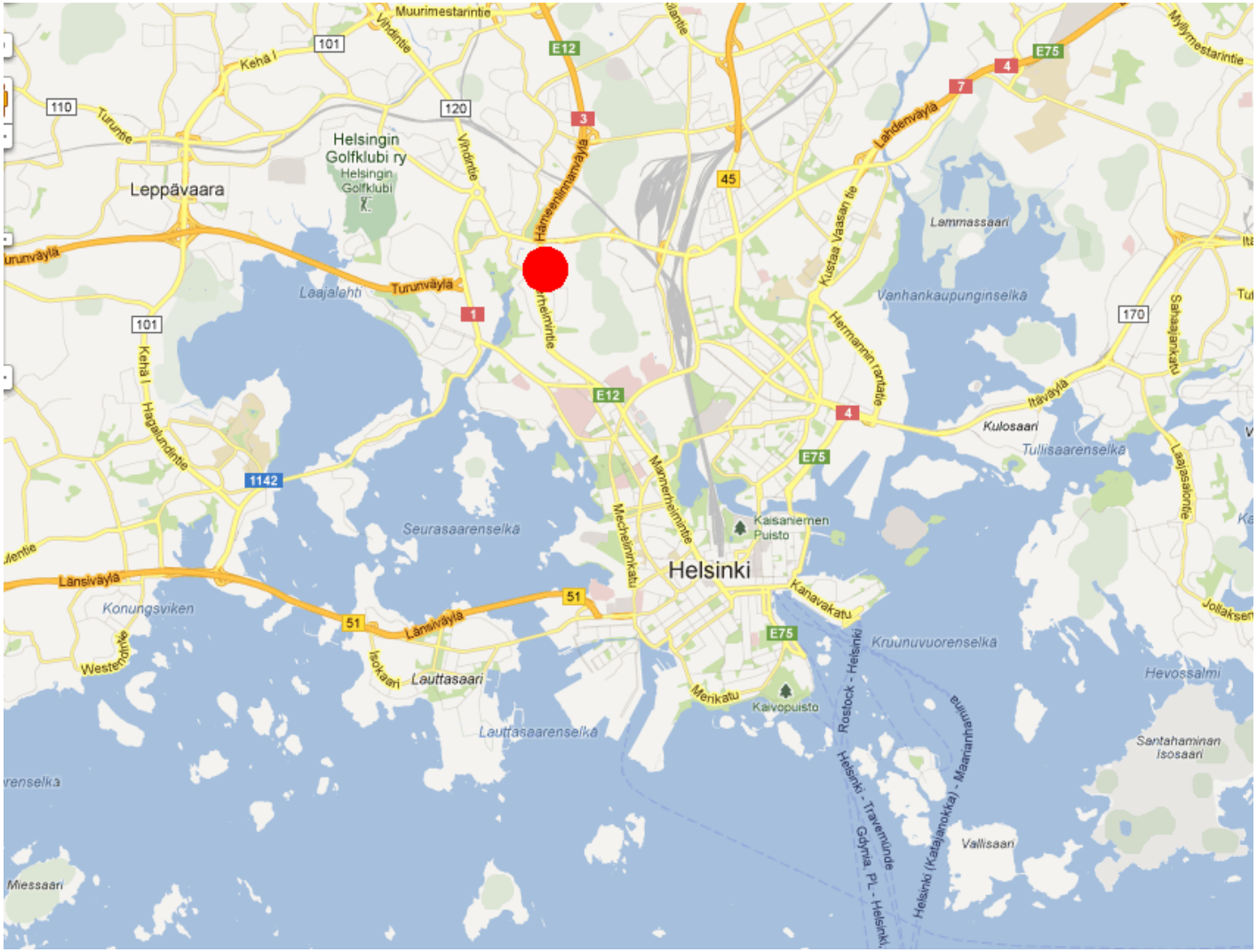
- Käytännössä, esim. kuljetuspalveluissa, reitinlaskentaongelma on usein monimutkaisempi
- Rajoituksia
  - Kapasiteetti - Ajoneuvoihin mahtuu vain tietty määrä tavaraa/ matkustajia kerrallaan
  - Aika - Kuljetus ei saa kestää liian kauan
  - Edeltävyys - Esim. noutopisteessä pitää käydä ennen toimituspistettä
- Tavoitefunktio: Lyhin reitti ei välttämättä ole paras
- Ajoneuvoja eli laskettavia reittejä voi olla useita

Reitinlaskenta kysyntäohjautuvassa joukkoliikenteessä

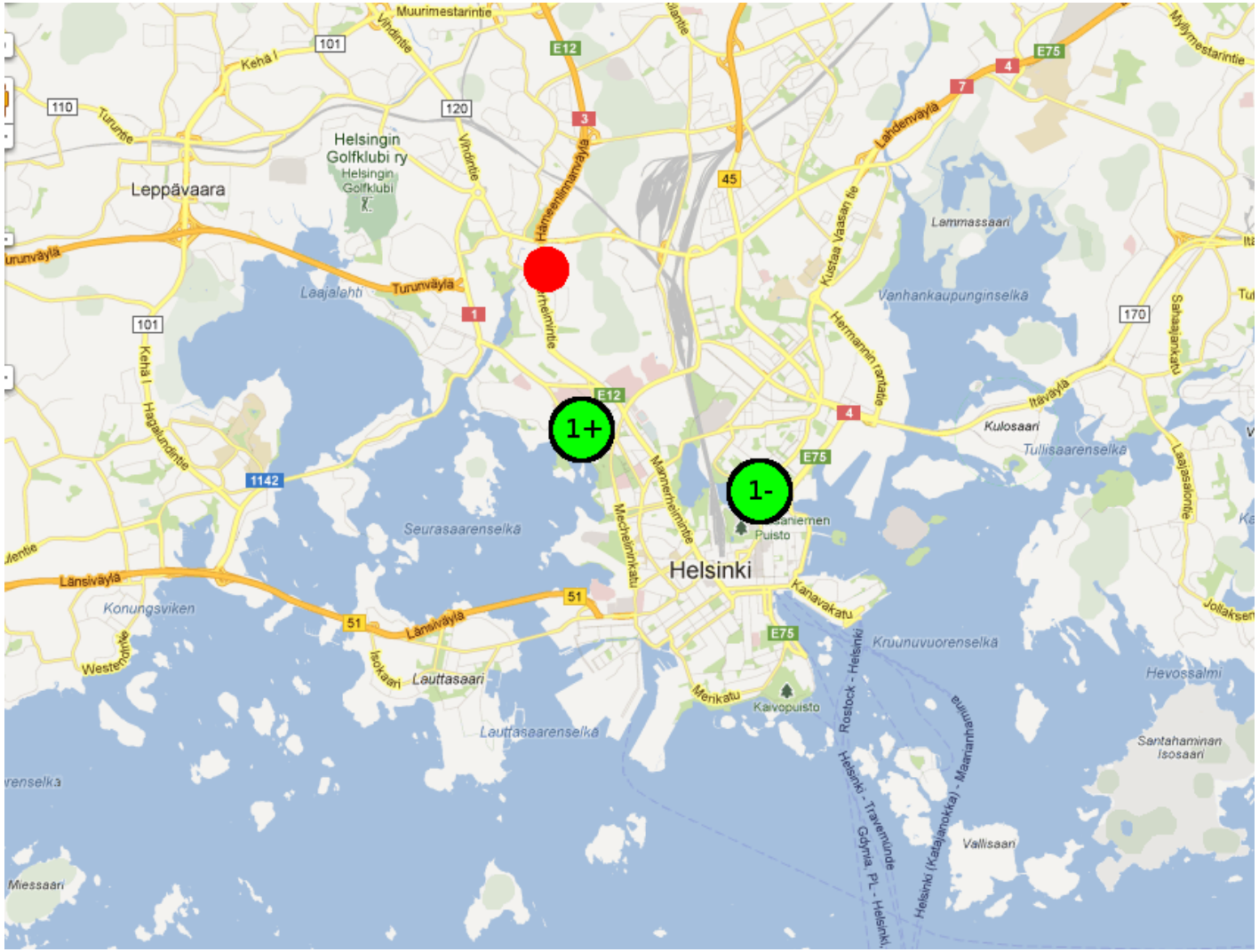
- Kysyntäohjautuva joukkoliikenne perustuu pienten tai keskisuurten ajoneuvojen (esim. minibussien) joustavaan reititykseen
- Asiakkaat voivat tilata matkoja reaaliaikaisesti esim. internet-käyttöliittymällä
- Ajoneuvojen reitit muodostuvat tilattujen matkojen perusteella
- Kunkin matkatilauksen yhteydessä ratkaistaan reitinlaskentaongelma



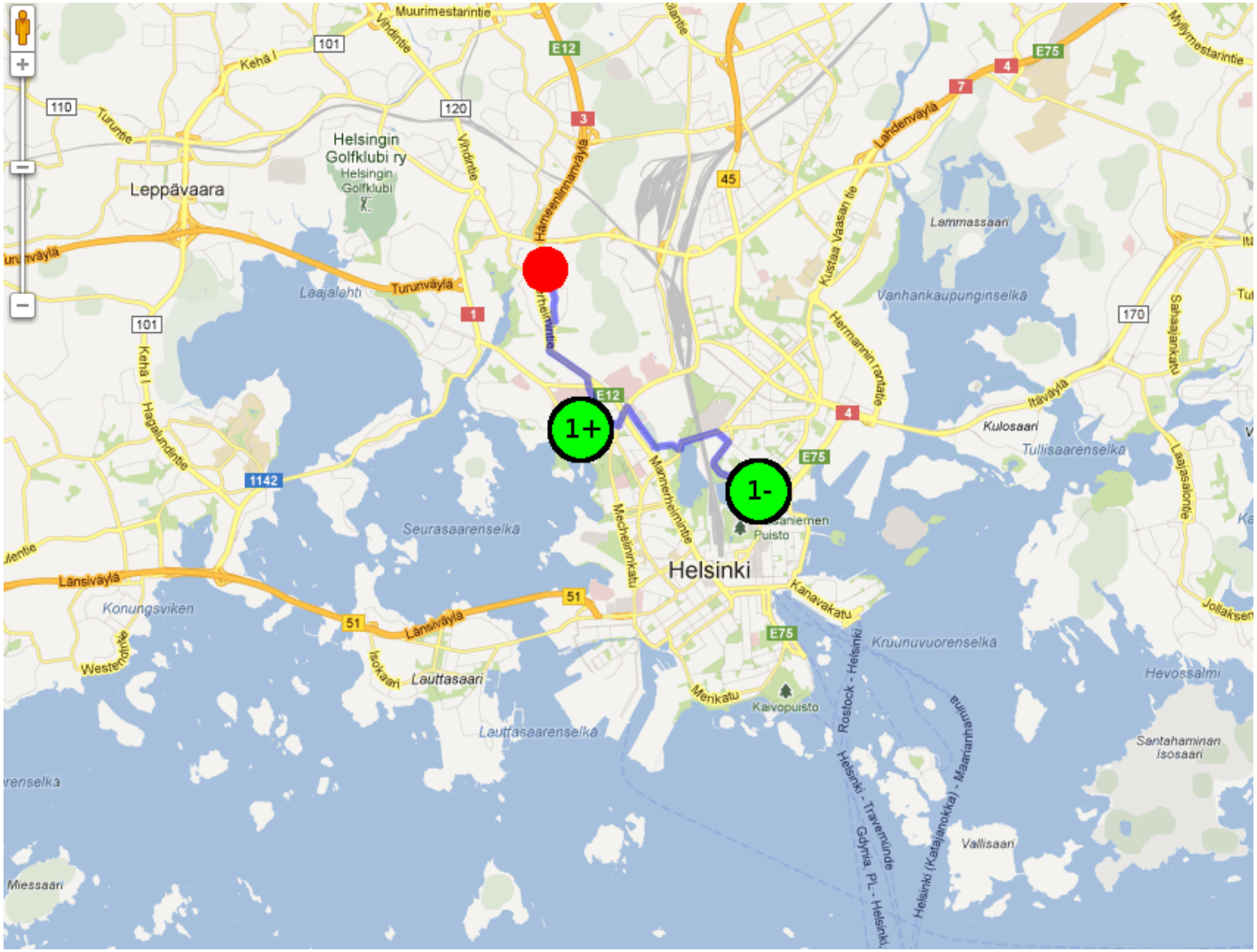
Kysyntäohjautuva joukkoliikenne, 1 ajoneuvo, esimerkki



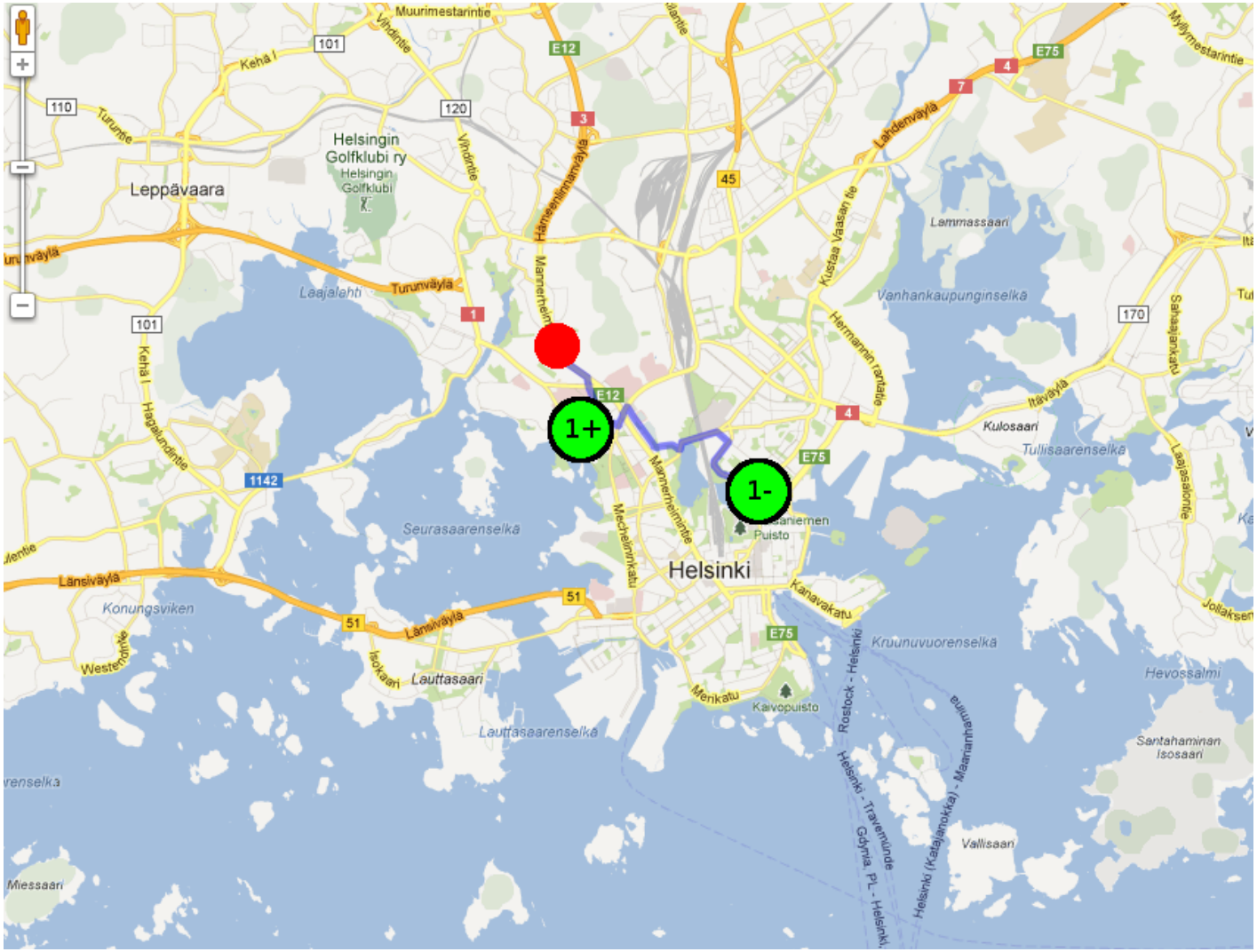
Kysyntäohjautuva joukkoliikenne, 1 ajoneuvo, esimerkki



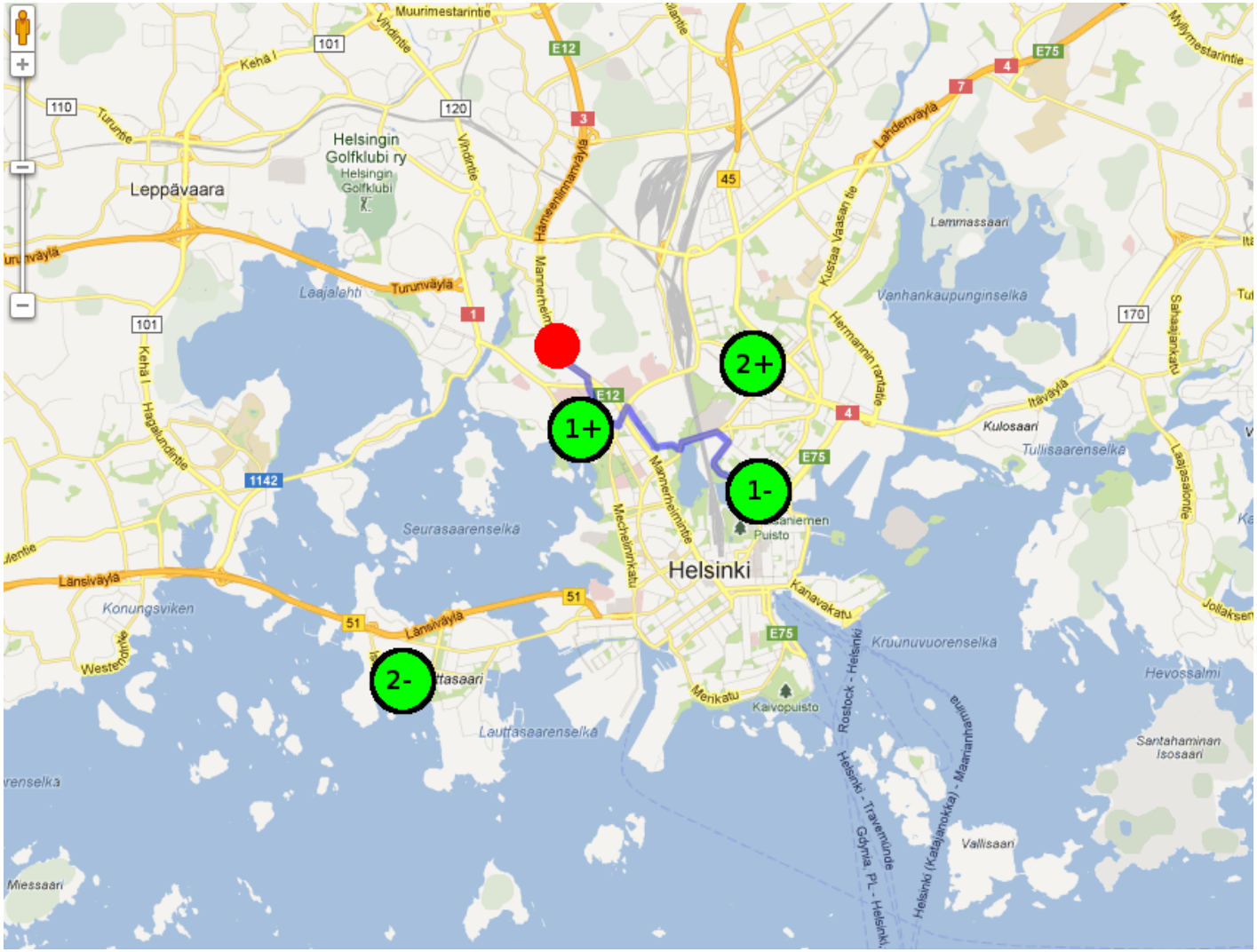
Kysyntäohjautuva joukkoliikenne, 1 ajoneuvo, esimerkki



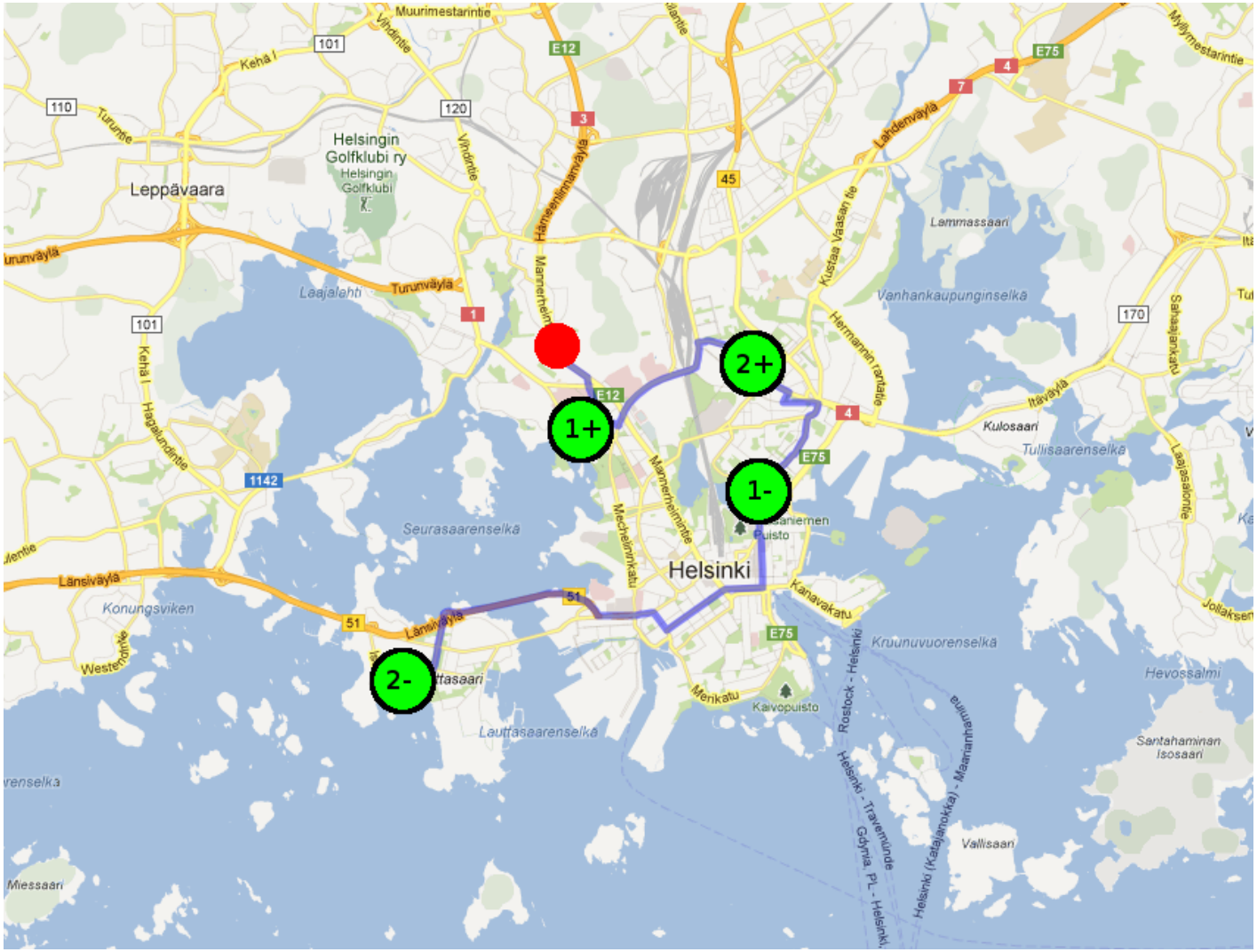
Kysyntäohjautuva joukkoliikenne, 1 ajoneuvo, esimerkki



Kysyntäohjautuva joukkoliikenne, 1 ajoneuvo, esimerkki



Kysyntäohjautuva joukkoliikenne, 1 ajoneuvo, esimerkki



Aikaikkunat

- Matka-aika voi pidentyä yllättäen reittimuutosten johdosta
- Palvelutasosta voidaan huolehtia ns. *aikaikkunoilla*
  - Esim. lähtö aikaisintaan klo 12:00, perillä viimeistään klo 13:00.
  - Aikaikkunat voivat olla osittain asiakkaan ja osittain järjestelmän määrittämiä
- Aikaikkunoita käytetään reitinlaskennassa, jotta tietty minimipalvelutaso toteutuisi
- Liian tiukat aikaikkunat vähentävät reitin joustavuutta

Ajoneuvon ja reitin valintaongelma

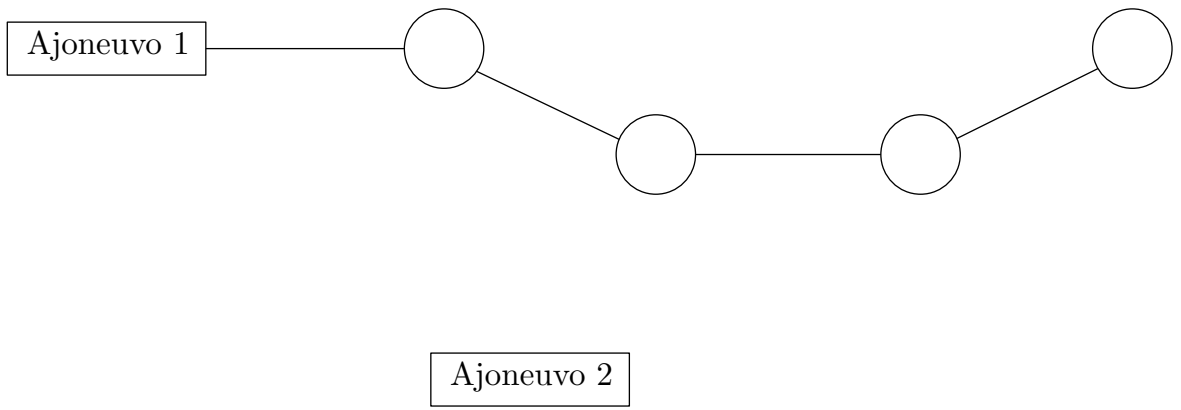
- Usean ajoneuvon tapauksessa jokaiselle uudelle asiakkaalle valitaan ajoneuvo ja valitulle ajoneuvolle määrätään uusi reitti
- Ajoneuvon ja reitin valinnassa pitää ottaa huomioon mm.
  - Uuden asiakkaan aiheuttama reitin pitenemä
  - Uuden asiakkaan palvelutaso ja muille asiakkaille aiheutuva palvelutason muutos
  - Kysyntäennuste
- Yleisesti jos minimoidaan reitin pituutta, palvelutaso saattaa kärsiä ja jos optimoidaan ainoastaan palvelutasoa, kustannukset kasvavat

Hajautettu ratkaisu

- Yritetään lisätä uusi asiakas johonkin olemassaolevista reiteistä
- Lasketaan jokaiselle ajoneuvolle uusi reittiehdotus ja valitaan niistä paras/parhaat
- Ajoneuvojen reittiehdotukset lasketaan erikseen

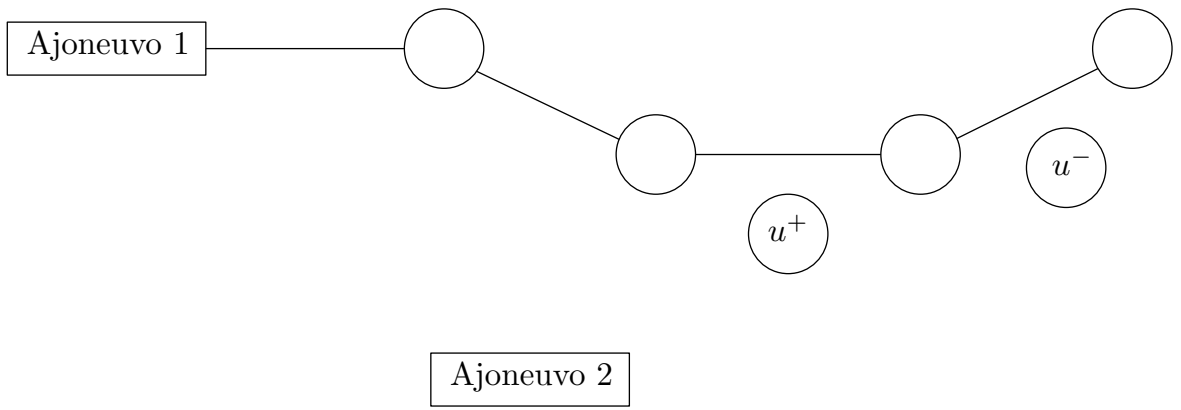
Hajautettu ratkaisu, esimerkki

- Kaksi ajoneuvoa, joista toinen odottaa tyhjänä



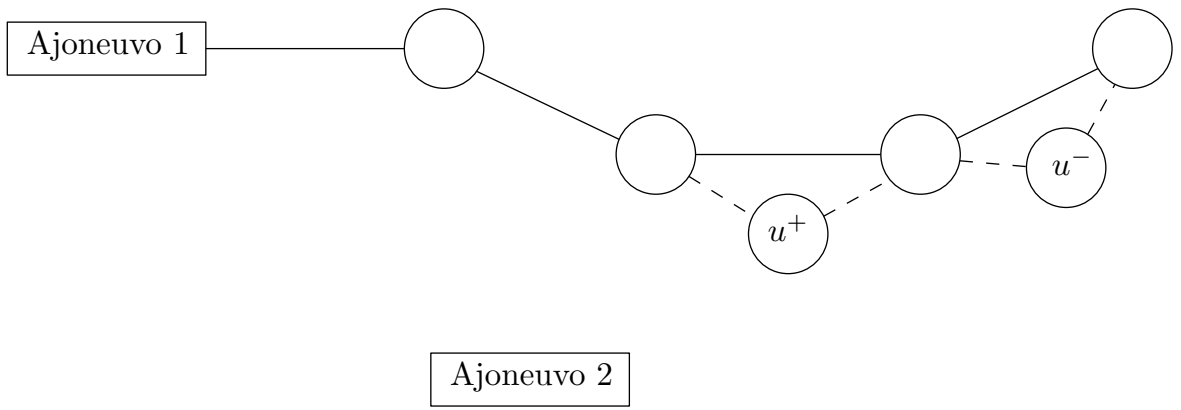
Hajautettu ratkaisu, esimerkki

- Kaksi ajoneuvoa, joista toinen odottaa tyhjänä
- Uusi asiakas tilaa matkan (u^+, u^-)



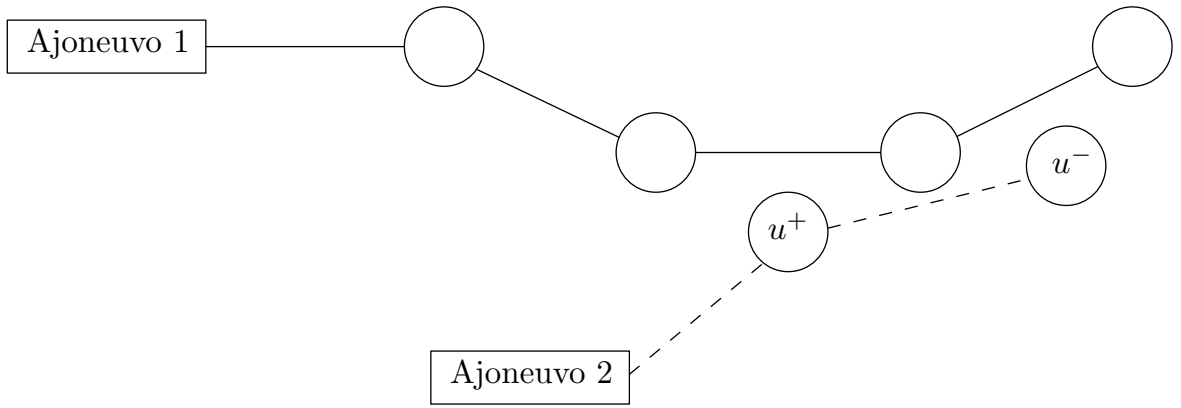
Hajautettu ratkaisu, esimerkki

- Kaksi ajoneuvoa, joista toinen odottaa tyhjänä
- Uusi asiakas tilaa matkan (u^+, u^-)
- Ehdotus 1: reitin pitenemä minimoituu, palvelutaso kärsii



Hajautettu ratkaisu, esimerkki

- Kaksi ajoneuvoa, joista toinen odottaa tyhjänä
- Uusi asiakas tilaa matkan (u^+, u^-)
- Ehdotus 1: reitin pitenemä minimoituu, palvelutaso kärsii
- Ehdotus 2: palvelutaso on paras mahdollinen, reitin pituus kasvaa enemmän

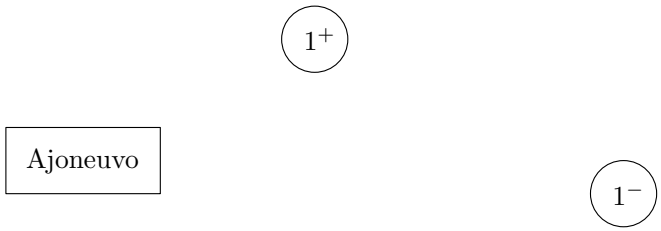


Yksinkertainen lisäysalgoritmi (Insertion algorithm)

- Yksinkertainen ratkaisu reittiehdotusten laskemiselle on lisätä uuden asiakkaan nouto- ja toimituspiste sopivaan väliin
- Ei-täydellinen ratkaisu: olemassaolevien pisteiden järjestys säilyy

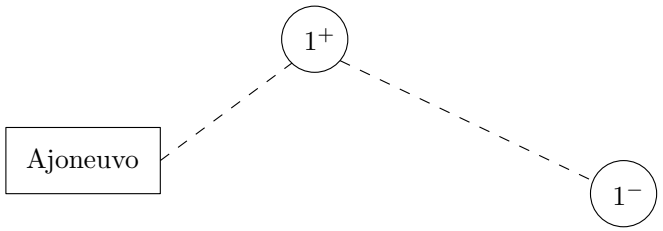
Yksinkertainen lisäysalgoritmi (Insertion algorithm)

- Yksinkertainen ratkaisu reittiehdotusten laskemiselle on lisätä uuden asiakkaan nouto- ja toimituspiste sopivaan väliin
- Ei-täydellinen ratkaisu: olemassaolevien pisteiden järjestys säilyy



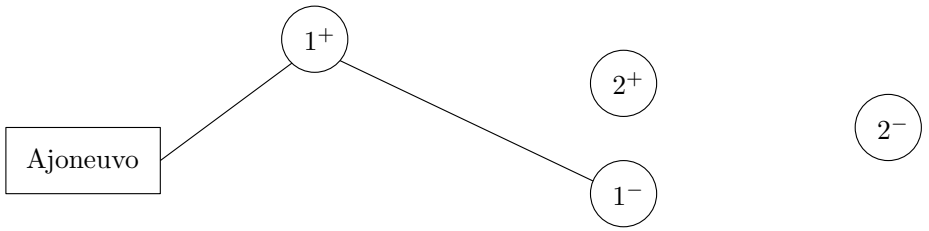
Yksinkertainen lisäysalgoritmi (Insertion algorithm)

- Yksinkertainen ratkaisu reittiehdotusten laskemiselle on lisätä uuden asiakkaan nouto- ja toimituspiste sopivaan väliin
- Ei-täydellinen ratkaisu: olemassaolevien pisteiden järjestys säilyy



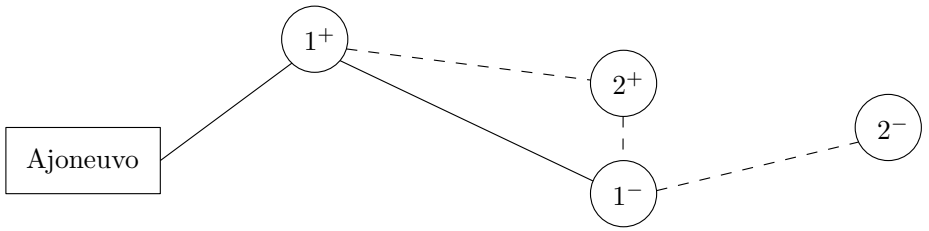
Yksinkertainen lisäysalgoritmi (Insertion algorithm)

- Yksinkertainen ratkaisu reittiehdotusten laskemiselle on lisätä uuden asiakkaan nouto- ja toimituspiste sopivaan väliin
- Ei-täydellinen ratkaisu: olemassaolevien pisteiden järjestys säilyy



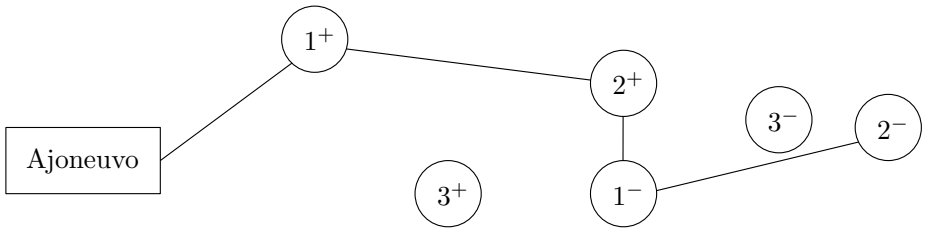
Yksinkertainen lisäysalgoritmi (Insertion algorithm)

- Yksinkertainen ratkaisu reittiehdotusten laskemiselle on lisätä uuden asiakkaan nouto- ja toimituspiste sopivaan väliin
- Ei-täydellinen ratkaisu: olemassaolevien pisteiden järjestys säilyy



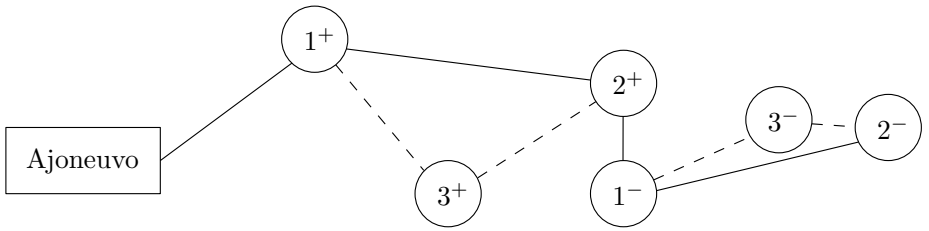
Yksinkertainen lisäysalgoritmi (Insertion algorithm)

- Yksinkertainen ratkaisu reittiehdotusten laskemiselle on lisätä uuden asiakkaan nouto- ja toimituspiste sopivaan väliin
- Ei-täydellinen ratkaisu: olemassaolevien pisteiden järjestys säilyy



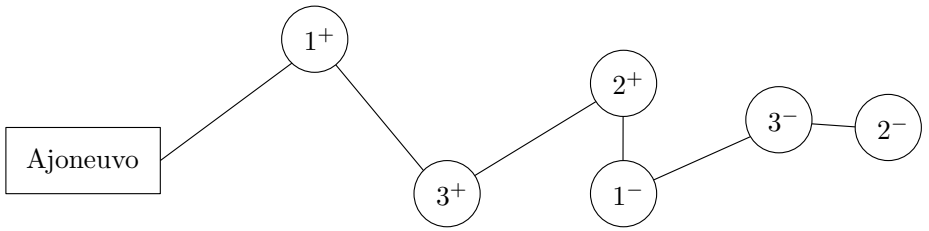
Yksinkertainen lisäysalgoritmi (Insertion algorithm)

- Yksinkertainen ratkaisu reittiehdotusten laskemiselle on lisätä uuden asiakkaan nouto- ja toimituspiste sopivaan väliin
- Ei-täydellinen ratkaisu: olemassaolevien pisteiden järjestys säilyy



Yksinkertainen lisäysalgoritmi (Insertion algorithm)

- Yksinkertainen ratkaisu reittiehdotusten laskemiselle on lisätä uuden asiakkaan nouto- ja toimituspiste sopivaan väliin
- Ei-täydellinen ratkaisu: olemassaolevien pisteiden järjestys säilyy



Laajennettu lisäysalgoritmi (Adaptive insertion algorithm)

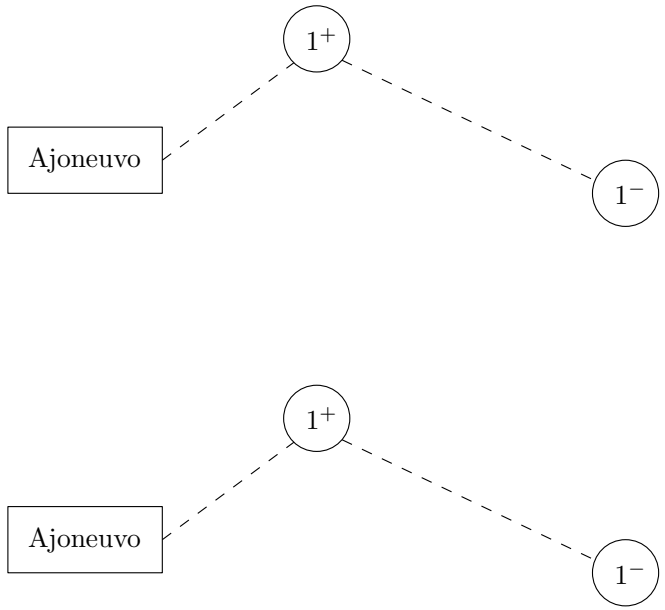
- Rakennetaan lisäysperiaatteella rinnakkain useampi vaihoehtoinen reitti ja valitaan niistä paras

Laajennettu lisäysalgoritmi (Adaptive insertion algorithm)

- Rakennetaan lisäysperiaatteella rinnakkain useampi vaihoehtoinen reitti ja valitaan niistä paras

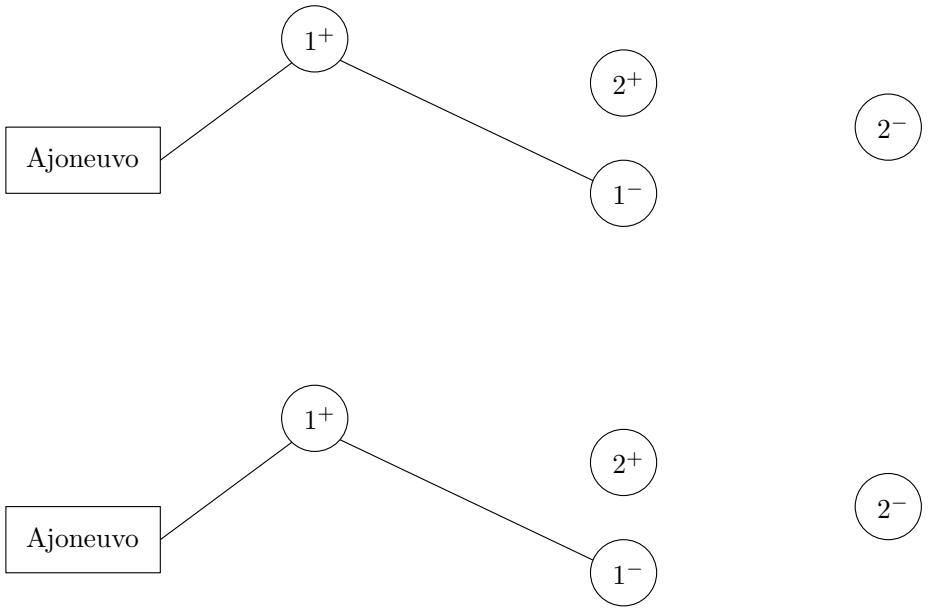
Laajennettu lisäysalgoritmi (Adaptive insertion algorithm)

- Rakennetaan lisäysperiaatteella rinnakkain useampi vaihoehtoinen reitti ja valitaan niistä paras



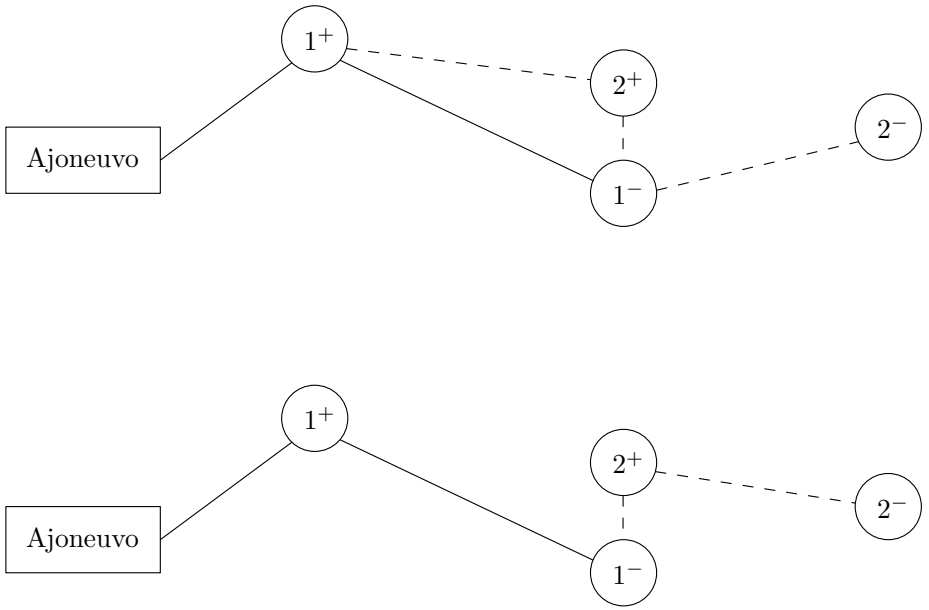
Laajennettu lisäysalgoritmi (Adaptive insertion algorithm)

- Rakennetaan lisäysperiaatteella rinnakkain useampi vaihoehtoinen reitti ja valitaan niistä paras



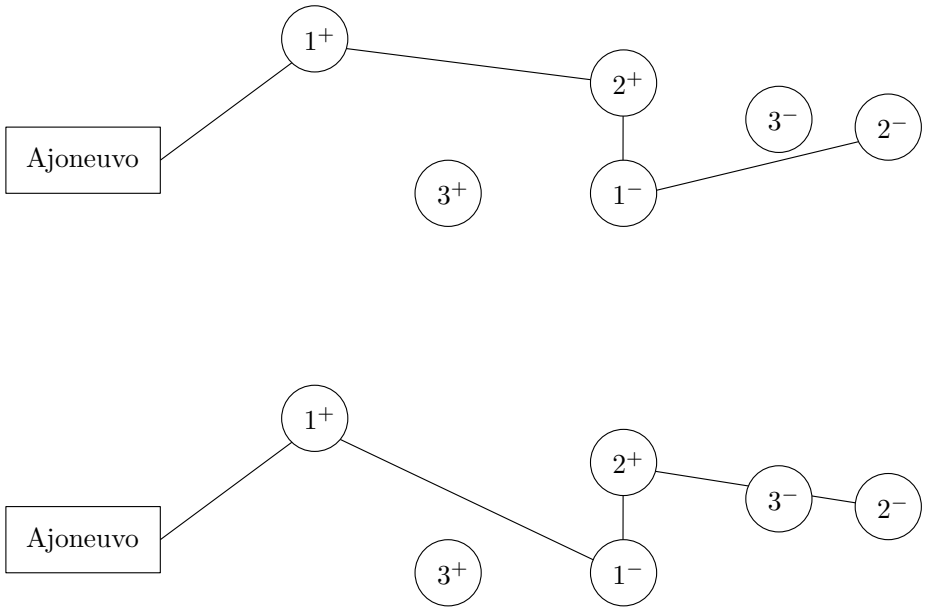
Laajennettu lisäysalgoritmi (Adaptive insertion algorithm)

- Rakennetaan lisäysperiaatteella rinnakkain useampi vaihoehtoinen reitti ja valitaan niistä paras



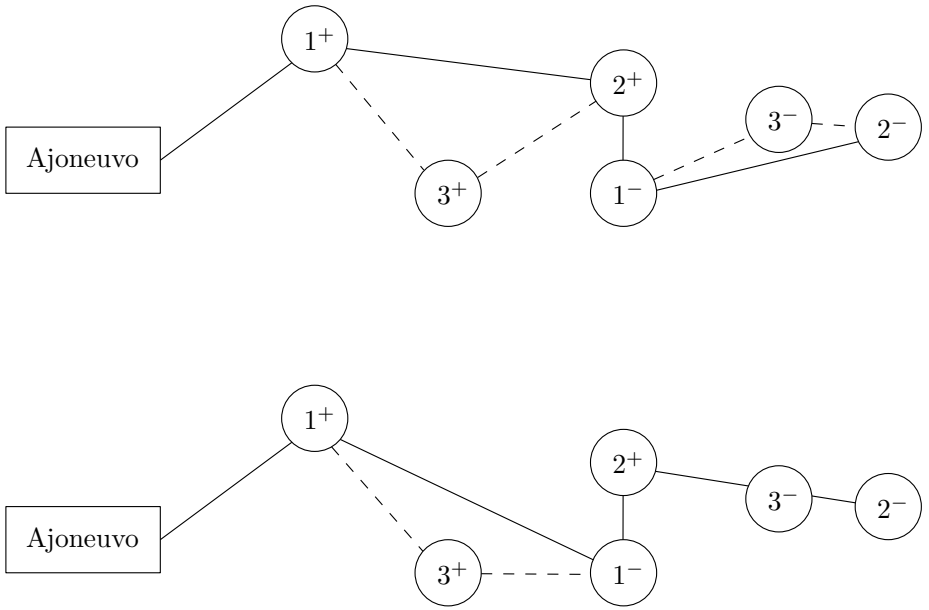
Laajennettu lisäysalgoritmi (Adaptive insertion algorithm)

- Rakennetaan lisäysperiaatteella rinnakkain useampi vaihoehtoinen reitti ja valitaan niistä paras



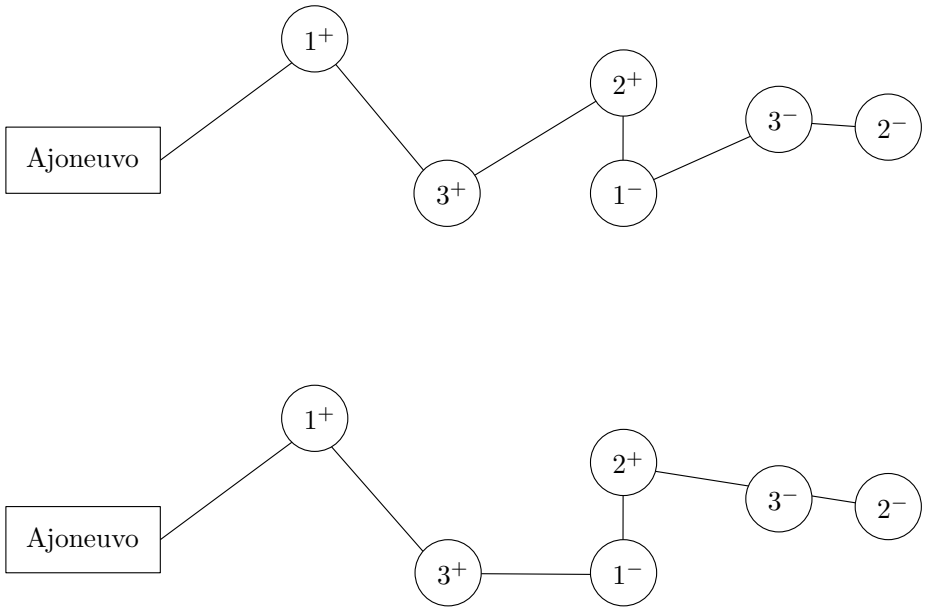
Laajennettu lisäysalgoritmi (Adaptive insertion algorithm)

- Rakennetaan lisäysperiaatteella rinnakkain useampi vaihoehtoinen reitti ja valitaan niistä paras



Laajennettu lisäysalgoritmi (Adaptive insertion algorithm)

- Rakennetaan lisäysperiaatteella rinnakkain useampi vaihoehtoinen reitti ja valitaan niistä paras



Täydellinen lisäysalgoritmi (Exact insertion algorithm)

- Rakennetaan lisäysperiaatteella rinnakkain kaikki mahdolliset reitit (enintään $\frac{(2n)!}{2^n}$ kpl)

$1^+, 1^-$

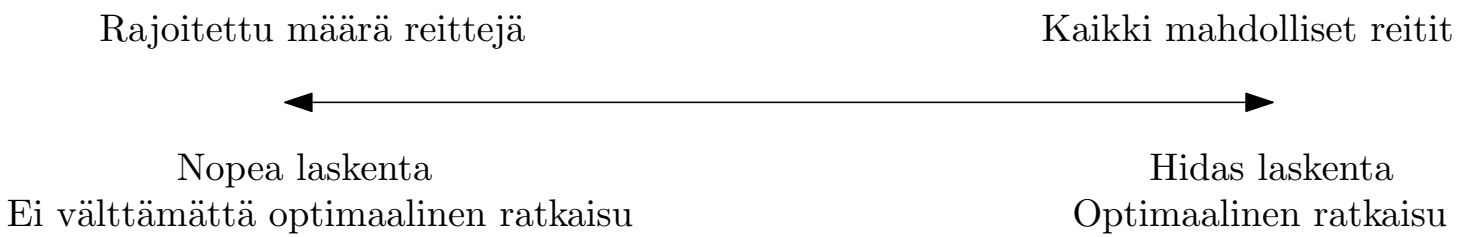
$1^+, 1^-, 2^+, 2^-$
$1^+, 2^+, 1^-, 2^-$
$1^+, 2^+, 2^-, 1^-$
$2^+, 1^+, 1^-, 2^-$
$2^+, 1^+, 2^-, 1^-$
$2^+, 2^-, 1^+, 1^-$

$1^+, 1^-, 2^+, 2^-, 3^+, 3^-$
$1^+, 1^-, 2^+, 3^+, 2^-, 3^-$
$1^+, 1^-, 3^+, 2^+, 2^-, 3^-$
$1^+, 3^+, 1^-, 2^+, 2^-, 3^-$
$3^+, 1^+, 1^-, 2^+, 2^-, 3^-$
$1^+, 1^-, 2^+, 3^+, 3^-, 2^-$
$1^+, 1^-, 3^+, 2^+, 3^-, 2^-$
$1^+, 3^+, 1^-, 2^+, 3^-, 2^-$
$3^+, 1^+, 1^-, 2^+, 3^-, 2^-$
$1^+, 1^-, 3^+, 3^-, 2^+, 2^-$
$1^+, 3^+, 1^-, 3^-, 2^+, 2^-$
$3^+, 1^+, 1^-, 3^-, 2^+, 2^-$

$\ldots$

Lisäysalgoritmi, tuloksia

- Tiukkojen aika- tai kapasiteettirajoitusten vallitessa kaikkien mahdollisten reittien lukumäärä pysyy kohtuullisena ja täydellinen algoritmi tuottaa nopeasti optimaalisen ratkaisun
- Jos rajoitukset eivät ole tiukkoja, saadaan tehokas ratkaisu rajoittamalla rinnakkaisten reittien lukumäärää

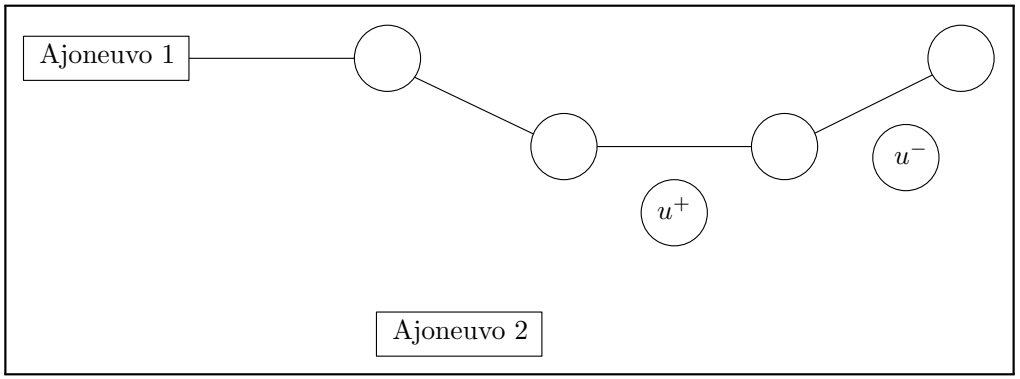


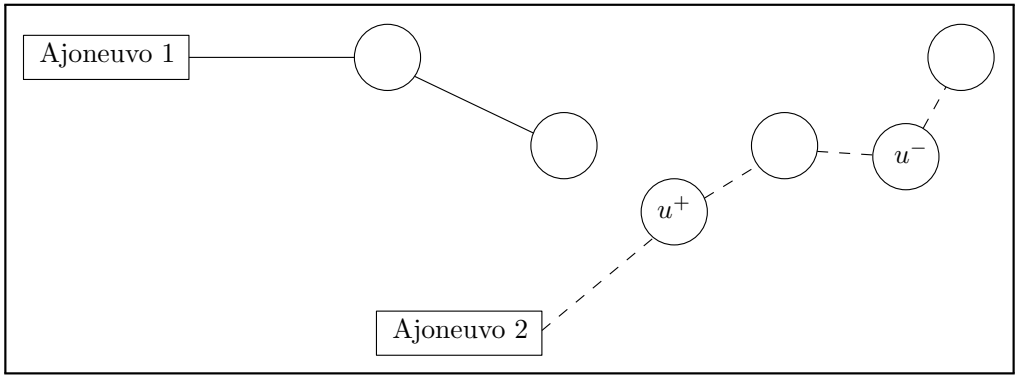
- Yksinkertainen lisäysalgoritmi on täydellinen, kun $n < 3$

Keskitetty ratkaisu

- Uuden matkatilauksen saapuessa etsitään parasta mahdollista asiakkaiden, ajoneuvojen ja reittien yhdistelmää
- Toistaiseksi noutamattomien asiakkaiden ajoneuvo voi vaihtua
- Periaate sisältää hajautetut ratkaisut

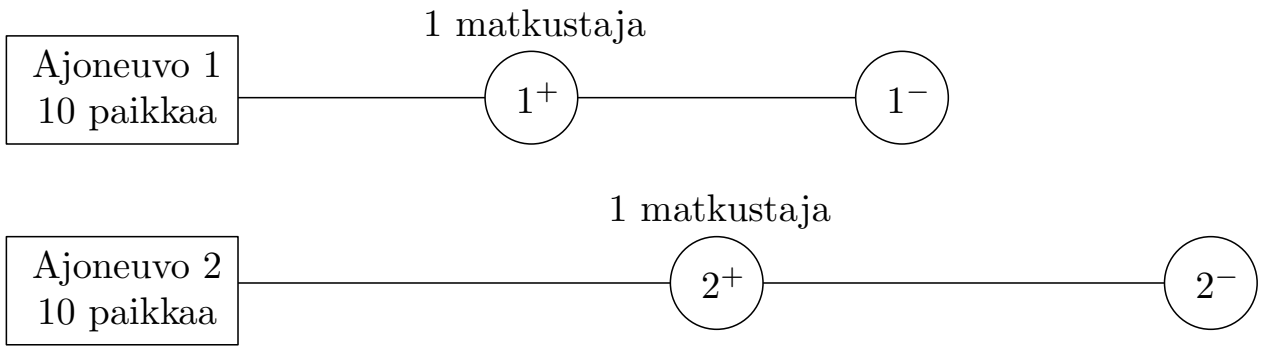
Keskitetty ratkaisu, esimerkki 1





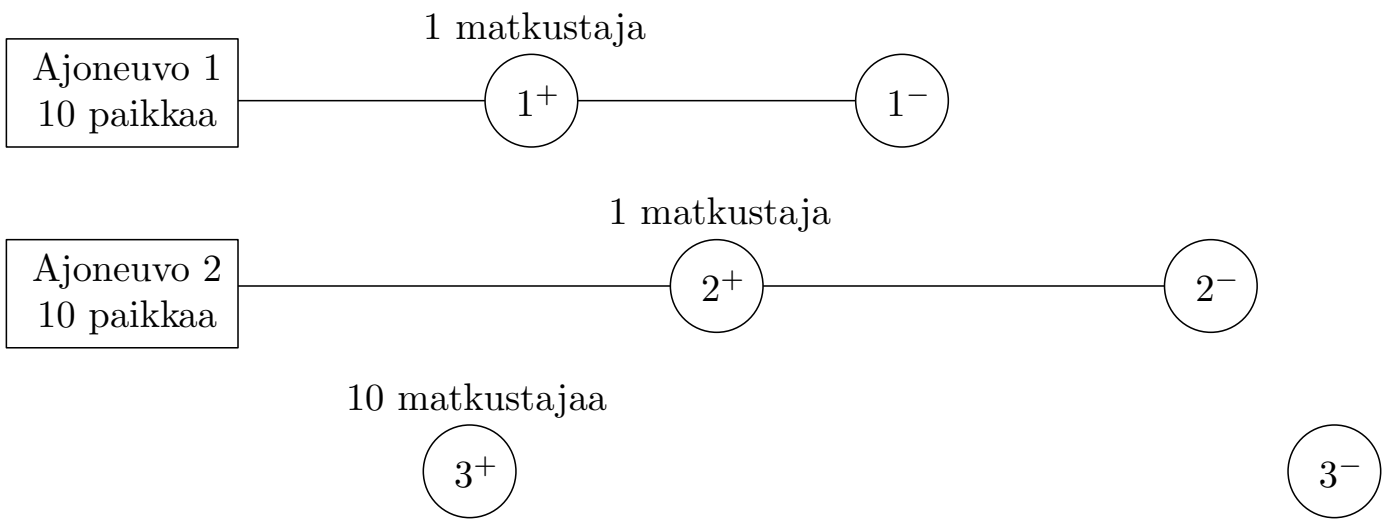
Keskitetty ratkaisu, esimerkki 2

- Keskitetyn ratkaisun merkitys korostuu rajoitetuissa tapauksissa



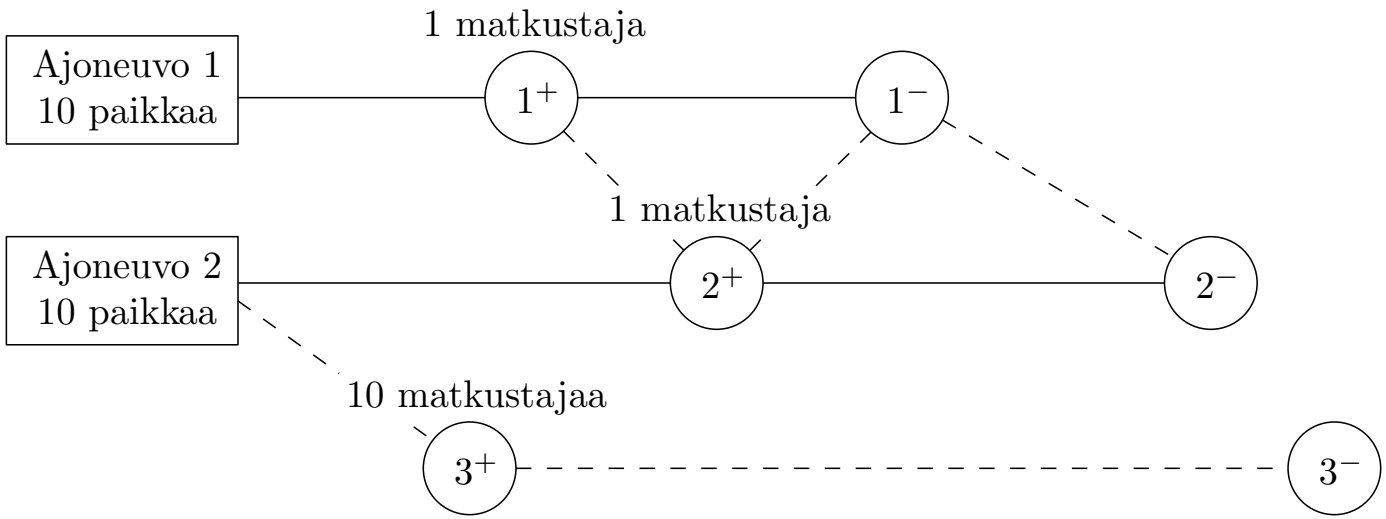
Keskitetty ratkaisu, esimerkki 2

- Keskitetyn ratkaisun merkitys korostuu rajoitetuissa tapauksissa



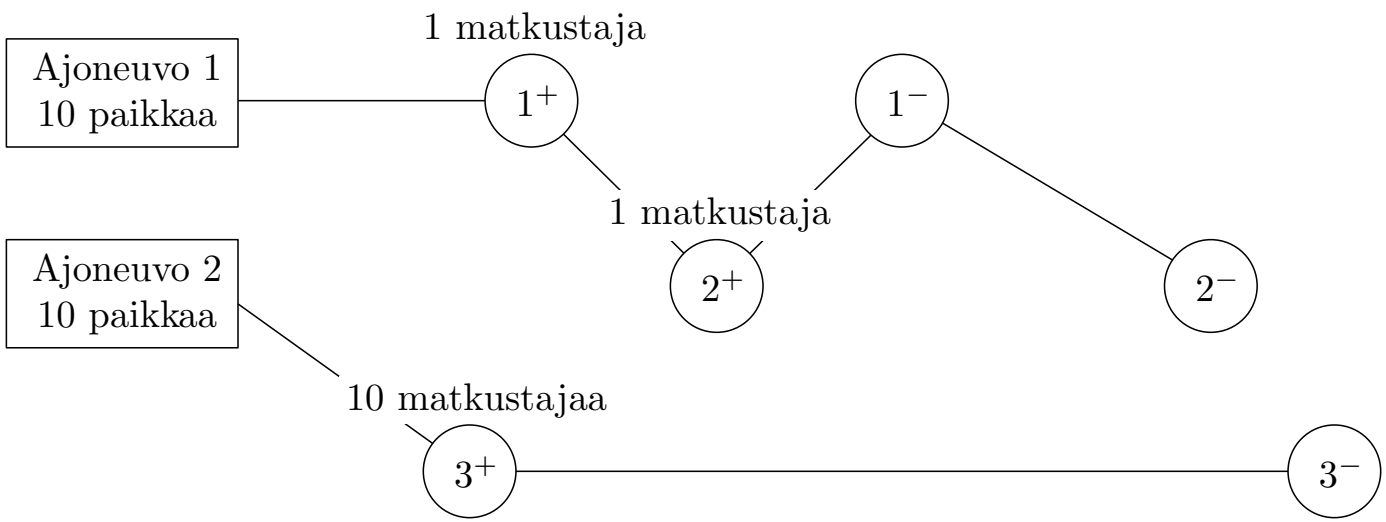
Keskitetty ratkaisu, esimerkki 2

- Keskitetyn ratkaisun merkitys korostuu rajoitetuissa tapauksissa



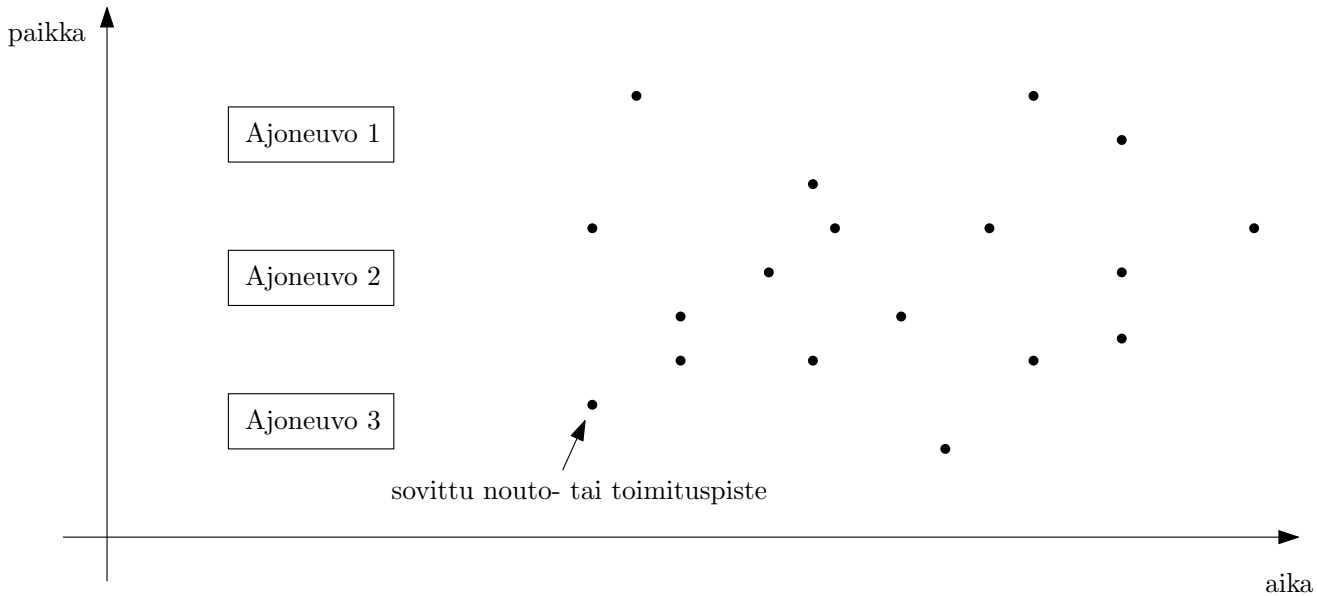
Keskitetty ratkaisu, esimerkki 2

- Keskitetyn ratkaisun merkitys korostuu rajoitetuissa tapauksissa



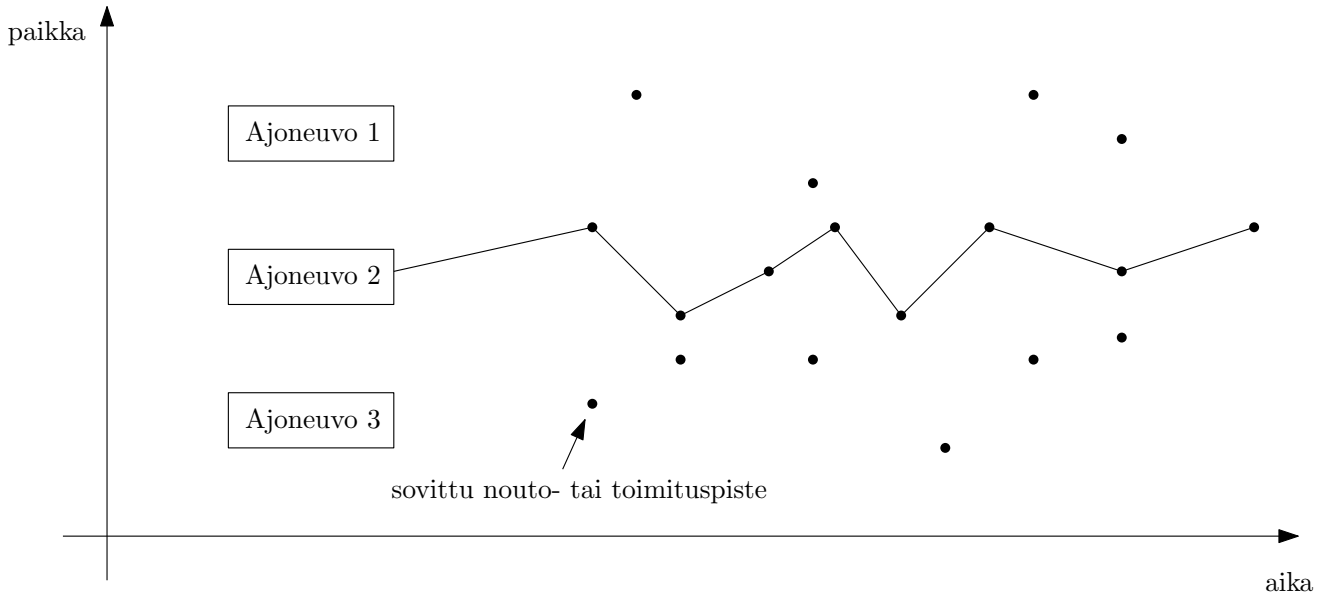
Maksimiklusteriperiaate (Maximum cluster algorithm)

- Perusidea: Pyritään palvelemaan mahdollisimman suuri asiakasjoukko (klusteri) aikarajojen sisällä kullakin ajoneuvolla
- Uuden matkatilauksen saapuessa klusterit lasketaan uudelleen



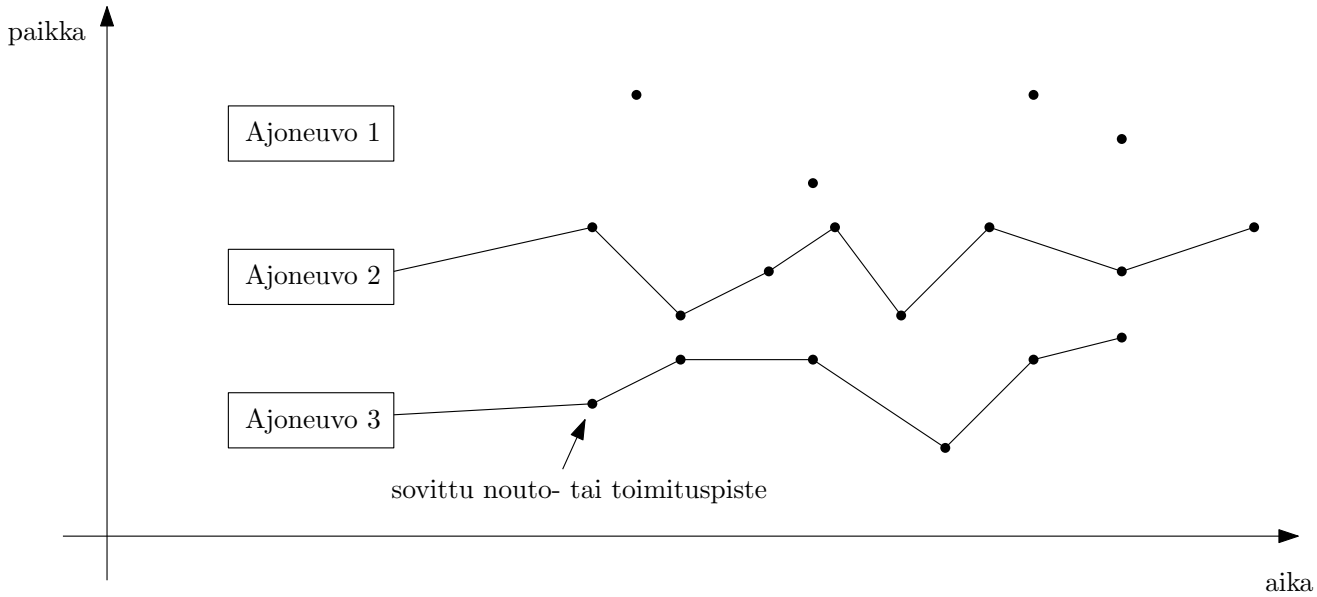
Maksimiklusteriperiaate (Maximum cluster algorithm)

- Perusidea: Pyritään palvelemaan mahdollisimman suuri asiakasjoukko (klusteri) aikarajojen sisällä kullakin ajoneuvolla
- Uuden matkatilauksen saapuessa klusterit lasketaan uudelleen



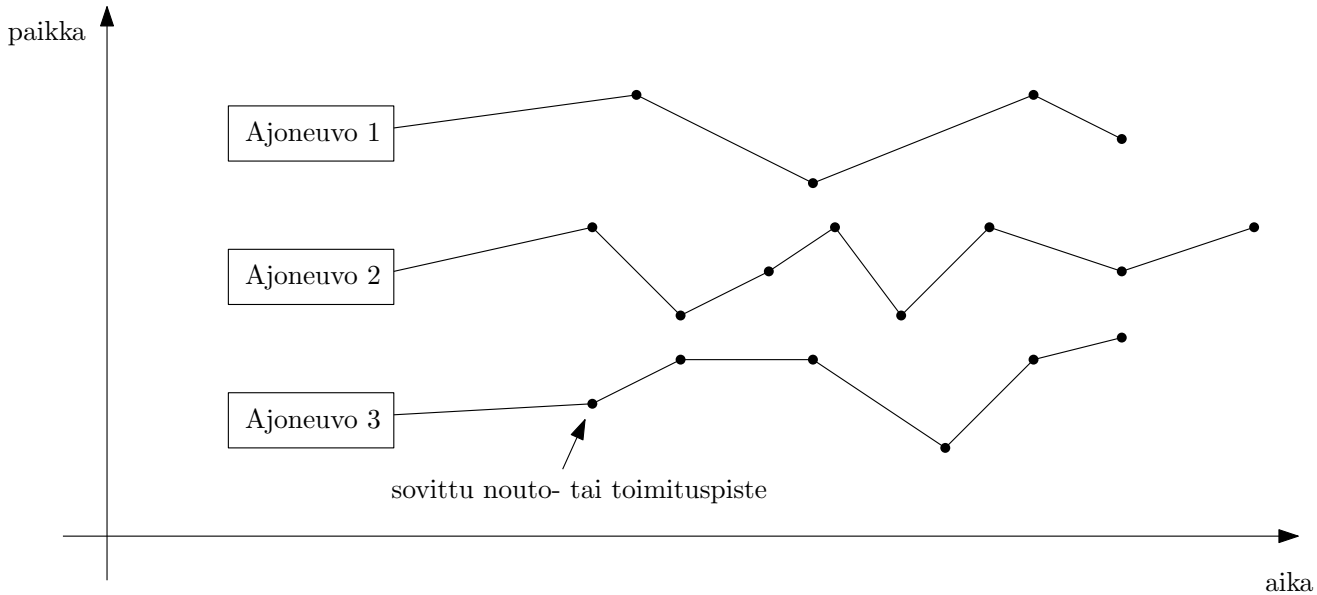
Maksimiklusteriperiaate (Maximum cluster algorithm)

- Perusidea: Pyritään palvelemaan mahdollisimman suuri asiakasjoukko (klusteri) aikarajojen sisällä kullakin ajoneuvolla
- Uuden matkatilauksen saapuessa klusterit lasketaan uudelleen



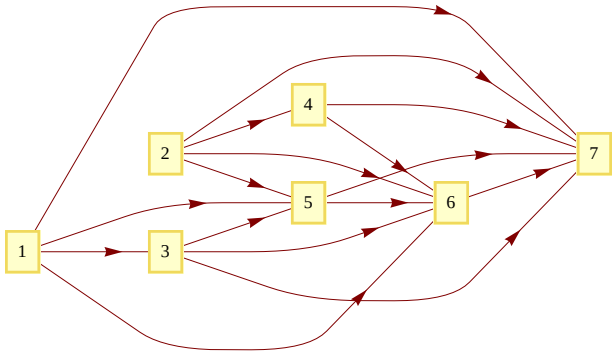
Maksimiklusteriperiaate (Maximum cluster algorithm)

- Perusidea: Pyritään palvelemaan mahdollisimman suuri asiakasjoukko (klusteri) aikarajojen sisällä kullakin ajoneuvolla
- Uuden matkatilauksen saapuessa klusterit lasketaan uudelleen



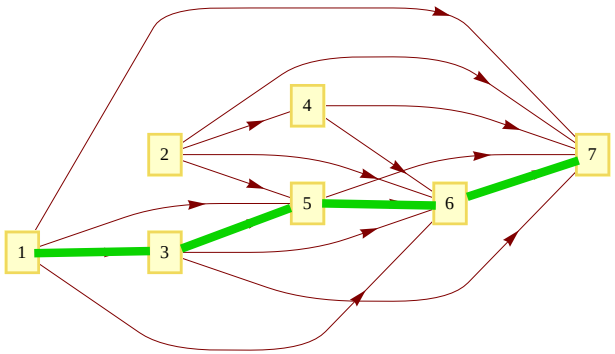
Arvojärjestysalgoritmi (Ranking algorithm)

- Maksimiklusterit voidaan määrittää tehokkaasti järjestämällä nouto- ja toimituspisteet arvojärjestykseen
- Suurimman arvon saavat pisteet, joista on mahdollista siirtyä mahdollisimman moneen arvokkaaseen pisteeseen aikarajojen sisällä
- Arvojärjestys saadaan laskemalla suurinta ominaisarvoa vastaava ominaisvektori (ks. HITS-hakualgoritmi)



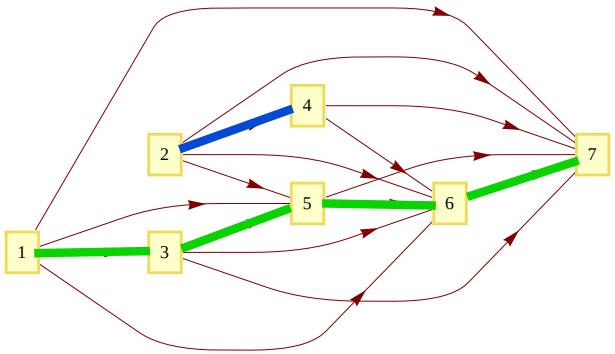
Arvojärjestysalgoritmi (Ranking algorithm)

- Maksimiklusterit voidaan määrittää tehokkaasti järjestämällä nouto- ja toimituspisteet arvojärjestykseen
- Suurimman arvon saavat pisteet, joista on mahdollista siirtyä mahdollisimman moneen arvokkaaseen pisteeseen aikarajojen sisällä
- Arvojärjestys saadaan laskemalla suurinta ominaisarvoa vastaava ominaisvektori (ks. HITS-hakualgoritmi)



Arvojärjestysalgoritmi (Ranking algorithm)

- Maksimiklusterit voidaan määrittää tehokkaasti järjestämällä nouto- ja toimituspisteet arvojärjestykseen
- Suurimman arvon saavat pisteet, joista on mahdollista siirtyä mahdollisimman moneen arvokkaaseen pisteeseen aikarajojen sisällä
- Arvojärjestys saadaan laskemalla suurinta ominaisarvoa vastaava ominaisvektori (ks. HITS-hakualgoritmi)



Arvojärjestysalgoritmi, tuloksia

- Arvojärjestysalgoritmi tuottaa tehokkaasti käypiä ratkaisuja tiukkojen rajoitusten vallitessa
- Kertaluokkaa nopeampi aikaisempiin menetelmiin verrattuna

Matkansuunnittelu

- Matkansuunnittelu (Journey planning) = joukkoliikennevälineen ja reitin valinta
- Tarkoituksena on löytää matkustajalle paras reitti ja aikataulu lähtöpisteestä määränpäähän, esim.
  - 16:27: kävely pysäkille A,
  - 16:39: bussi numero 58 pysäkiltä A pysäkille B
  - 16:53: kävely pysäkiltä B määränpäähän, perillä klo 17:11

Stokastinen malli

- Olemassaolevilla menetelmillä voidaan laskea etukäteen paras reitti esim. matka-ajan, odotusajan, kävelymatkan tai vaihtojen lukumäärän suhteen
- Todellisuudessa paras reitti ei välttämättä toteudu esim. myöhästymisien tai vuorojen peruutuksien takia
- Stokastinen malli ottaa huomioon mahdolliset reittimuutokset matkan varrella
- Mallin avulla voidaan laskea paras matkastrategia eri tavoitteiden suhteen