# Kvadratnet – afstand til nærmeste stoppested

Indhold

[Kvadratnet – afstand til nærmeste stoppested 1](#_Toc179894831)

[Formål 1](#_Toc179894832)

[Data 1](#_Toc179894833)

[Procedurebeskrivelse 2](#_Toc179894834)

[Dokumentation af løsning 3](#_Toc179894835)

[Effektivitet 3](#_Toc179894836)

[Backlog 3](#_Toc179894837)

## Formål

Beregn distancen fra hvert beboet område i Region Midtjylland til det nærmeste stoppested.

Kan bruges til at analysere i hvilke geografiske områder den kollektive trafik skal udvides.

## Data

* Befolkningskvadratnet for Region Midtjylland med geometry kolonne (som udgangspunkt 100m\*100m)
  + F:\Køreplanlægning\Data og Analyse\Grunddata\Befolkning\...\xxx.shp
  + Skal mindst indeholde kolonnerne: geometry og antal\_tal
* Nyeste udtræk af alle stoppesteder i Region Midtjylland med UTM32N koordinater (både bus, letbane, plustur, flextur)
  + F:\Køreplanlægning\Data og Analyse\Grundata\Stop\...\xxx.csv
  + Skal mindst indeholde kolonnerne: UTM32\_Easting, UTM32\_Northing, Long name og kode til stoppunkt
* Dobbeltrettet OSM netværk af typen ”walk”
  + Hentes automatisk af scriptet
  + ”walk” inkluderer alle veje og stier
  + ”all\_private” inkluderer alle veje og stier inklusiv private og service veje

## Procedurebeskrivelse

* Opdater mappen Data med nyeste befolkningskvadratnet for Region Midtjylland
* Opdater mappen Data med nyeste stoppunkter
* Åbn script og ret filnavne
* Evt. ret chunk size fra 500 til noget lavere hvis memory er et problem
* Kør script (ca. 30-45 minutter)
* Upload resultat til Webgis

Resultatet indeholder:

* Det originale kvadratnet
* Distancen mellem centroiden af kvadratet og gps punkt for nærmeste stander
* Stander navn og standernummer for nærmeste stander

## Dokumentation af løsning

Lavet med inspiration fra: [osmnx-examples/notebooks/14-osmnx-to-igraph.ipynb at main · gboeing/osmnx-examples · GitHub](https://github.com/gboeing/osmnx-examples/blob/main/notebooks/14-osmnx-to-igraph.ipynb)

Anvender OSMNX og NetworkX til at håndtere OSM netværk.

Anvender igraph, som er bygget i C, til super hurtige udregninger af Dijkstras algoritme.

1. Hent OSM netværk med OSMNX
2. Omdan OSM netværket til en igraph graf hvor kanter er vægtet med kantlængde i meter
3. Gem en mapping af igraph id til osmid og en mapping af osmid til igraph id
4. Find nærmeste OSM node til alle centroider på kvadratnettet og gem distancen
5. Find nærmeste OSM node til alle stop og gem distancen
6. Fjern stop hvis distancen mellem stop og nærmeste OSM node er > 1000 meter. Det betyder at stoppet er udenfor Midtjylland.
7. Oversæt OSM nodes til igraph nodes
8. Find korteste distance fra hver stop node til alle nodes på grafen
9. For hver centroide node, find det stop med kortest distance

### Effektivitet

Umiddelbart er problemet at finde distancen fra hver centroide til det nærmeste stoppested dvs. centroide 🡪 alle stop. Det er muligt at udregne fugleflugtsdistancen fra hver centroide til alle stops og kun udføre Dijkstras algoritme til de nærmeste K stop.

Problemet er dog, at man er nødt til at gentage beregningerne for hver centroide, og da mange centroider befinder sig tæt på hinanden og har (næsten) samme nærmeste stoppested og sti dertil, betyder det mange repetitive (overflødige) beregninger.

Problemet kan vendes om til stop 🡪 alle centroider. For at undgå repetitive beregninger kan problemet omformuleres til stop 🡪 alle nodes på grafen. Det udregner mange unødvendige distancer, dog skal alle beregninger kun udføres én gang og løser problemet stop 🡪 alle centroider da centroider er en delmængde af alle nodes.

Igraph er et Python interface til en C implementation af Dijkstras algoritme og den paralleliserer udregningerne til alle CPU-kerner via multiprocessing.

## Backlog

* Forkerte udregninger på Venø