Jorissen,Laurence L.M.C.B., 5507944

fontys  de rondom 1, 13-10-2025

verslag opdracht OR5

Contents

[Probleem definitie 2](#_Toc211510638)

[Input 2](#_Toc211510639)

[Beslisvariabelen 2](#_Toc211510640)

[Doelfunctie 2](#_Toc211510641)

[computationele complexiteit 2](#_Toc211510642)

[onderzoeks aanpak 3](#_Toc211510643)

[onderzoeksvraag 1: 3](#_Toc211510644)

[onderzoeksvraag 2: 3](#_Toc211510645)

# Probleem definitie

## Input

Als input is er een groot dataframe gelevert met 120 x- en y-coördinaten geleverd. Er staat ook nog een startpunt in, dus waar de krantenbezorgers hun kranten moeten ophalen. Om de opdracht te vereenvoudigen heb ik besloten om het grote dataframe in 4 kleinere dataframes op te splitsen. Dit heb ik opgesplitst in waar de coördinaten liggen. Alle coördinaten die rechtsboven liggen heb ik dus in een dataframe, dit heb ik voor alle 4 de dataframes zo gedaan.

## Beslisvariabelen

Nearest neighbor:

Het algoritme maakt een keuze voor het volgende punt op basis van de afstand tussen de punten. Het kiest voor het punt wat het dichtste bij ligt. Vervolgens zet het het gekozen punt in een lijst. Op deze manier wordt een punt niet dubbel gebruikt. Om dit in wiskundige notatie te zetten ziet dit er als volgt uit.

## Doelfunctie

Het doel van het algoritme is om de routes van de 4 krantenbezorgers te minimaliseren, waarbij ieder punt 1 keer wordt bezocht.

St.

## computationele complexiteit

# onderzoeks aanpak

## onderzoeksvraag 1:

**Hoe presteren jouw metaheuristiek en jouw constructieve heuristiek?**

Als constructieve heuristiek is er gekozen voor nearest neighbor (NN). NN is tijdsefficiënt wat een voordeel is, verder komt er altijd een oplossing uit. Dit zijn 2 grote voordelen bij de NN aanpak. Het nadeel van NN is echter dat het geen optimale oplossing is. Bij simulated annealing is het voordeel dat hij van de nearest neighbor oplossing hem optimaler kan maken. Bij simulated annealing kom je nog altijd niet uit op een globaal optimum, maar wel bij een lokaal optimum. Dit zorgt ervoor dat de oplossing al beter is dan de oplossing van nearest neighbor. Het nadeel van simulated annealing is echter dat het niet op zichzelf kan staan. Er is altijd al een oplossing nodig die correct is om vervolgens pas simulated annealing toe te kunnen passen.

## onderzoeksvraag 2:

**Welke parameter setting werken het beste voor je metaheuristiek? Leg ook uit hoe je dit hebt bepaald.**

Aangezien er gekozen is om het dataframe in 4 delen op te splitsen is er eerst gekozen om ze is 4 gelijken stukken te splitsen. Ieder dataframe had een gelijke oppervlakte. Hier kwam echter uit dat de krantenbezorgers voor het gedeelte van linksboven en rechtsonder een aanzienlijk langere route hadden dan linksonder en rechtsboven. Vervolgens is het meerdere keren aangepast om te zorgen dat de routes van alle krantenbezorgers ongeveer even lang zijn. De volgende parameters horen hierbij: de dataframes zijn gescheiden door een x-coördinaat van 275, dus de bovenste dataframes hebben alle x-coördinaten groter dan 275, en de onderste dataframes hebben x-coördinaten kleiner dan 275. De linker dataframes zijn gescheiden door een y-coördinaat van 310 en de rechter dataframes zijn gescheiden door een y-coördinaat van 250. Door deze parameters zijn de routes van iedere krantenbezorger ongeveer even lang.

## Onderzoeksvraag 3:

**Wat is het effect van onze gekozen startoplossing (random t.o.v. greedy) op de  
kwaliteit van de verkregen eindoplossing?**

# Algoritmisch design

## Rationele ontwerp keuzes

Om te beginnen is er gekozen om te beginnen met nearest neigbour, omdat het een snelle correcte oplossing biedt als eerste start voor het probleem. Het zorgt ervoor dat je een uitgangspositie hebt waardoor er later simulated annealing toegepast kan worden. Dit zorgt ervoor dat er snel doorgegaan kan worden naar een optimalere oplossig. Verder is er gekozen om het dataframe in 4 kleine dataframes te splitsen, dit is een keuze geweest, omdat de eerste oplossing door middel van nearest neighbor hierdoor optimaler wordt. Op deze manier wordt ervoor gezorgd dat er minder kans is dat de krantenbezorgers elkaar kruisen. Dit zorgt er dus voor dat metaheuristiek minder moeite hoeft te doen om tot een lokaal optimum te komen. Dit zorgt er ook weer voor dat de functies tijdefficiënt zijn.

## Pseudocode voor de algoritmes

**Nearest neighbor:**

1. Data inladen en bekijken