Autors: Toms Kārlis Bicāns
St.apl.nr: tb19019

MPD1 risinājums

1. Uzdevuma formulējums:

Paku piekādes dronu optimizēta maršruta meklēšana. (VRP - vehicle routing problem). Mērķis ir noteikt optimālu maršrutu vienam vai vairākiem droniem, lai tie piegādātu pakas vairākiem klientiem, ņemot vērā dronu tehniskos ierobežojumus un piegādes prasības. Uzdevumā ietilps 1 dronu stacija un n piegādes punkti. Visi droni sāk ceļu no šīs stacijas. Uzdevums ir panākt, lai katrs piegādes punkts tiktu apmeklēts 1 reizi un beigās droni atgrieztos stacijā.

Īss izdarītā apraksts:

Tika izveidota programma python programmēšanas valodā, kas spēj izpildīt 7 dažādus testa scenārijus ar mainīgiem klientu skaitiem. Katrā testa scenārijā tika testēta algoritma darbība 3 dažādos domēnos – nejauši izkliedēti punkti, 3 pilsētas, 7 pilsētas. Programmas rezultāti tiek glabāti `tests` mapē github repozitorijā. Konkrētie risinājumi arī ir serializāti failos, un tos ir iespējams ielādēt atmiņā, lai, piemēram, turpinātu optimizācijas darbu. Tika izmantoti paši vienkāršākie algoritmi (apkārtnes funkcija, domēns). Pašā risinājumā neparādās nekas saistīts ar droniem, bet gan tas ir vispārīgs optimālā maršruta meklēšanas uzdevums 1 transportlīdzeklim, 1 stacijai un `n` klientiem.

Autors: Toms Kārlis Bicāns
St.apl.nr: tb19019

2. Algoritma apraksts

Izvēlētais algoritms: Simulated annealing (SA)

Domēns

Stacijas pozīcija: Centrālā stacija ir definēta ar fiksētu koordinātu sistēmu (x, y).

Klientu pozīcijas: Klientu atrašanās vietas tiek ģenerētas dažādos variācijās, piemēram, vienkāršs izlīdzinājums, trīs pilsētu klasteri vai septiņu pilsētu klasteri.

Maršruti: Viena transportlīdzekļa maršruts sastāv no secīgu punktu (klientu) apmeklēšanas, sākot un beidzot vienā un tajā pašā stacijā.

Novērtēšana (izmaksu funkcija):

Mērķis: Minimizēt kopējo nobraukto attālumu, bet programma ir izveidota tā, ka var definēt vairākas izmaksu funkcijas un tās visas saskaitīt kopā, kā tika stāstīts lekcijā.

Aprēķins: Kopējais attālums tiek aprēķināts kā eiklīda attālums starp secīgi apmeklētajiem punktiem maršrutā.

Gājiens:

Gājiena veids: Nejauša divu klientu vietu maiņa maršrutā. Konkrēti, tiek izvēlēti divi nejauši klienti (izņemot sākuma un beigu punktus, kas ir stacija) un to vietas apmainītas. Ļoti primitīvs algoritms.

Autors: Toms Kārlis Bicāns
St.apl.nr: tb19019

3. Testēšanas apraksts

Testa piemēri

Tika izstrādāti vairāki testpiemēri ar mainīgu klientu skaitu un dažādām klientu izvietojuma variācijām:

Klientu skaits (Kopā 7 scenāriji):

- Mazs: 10 klienti
- Vidējs: 20, 30, 40, 50 klienti
- Liels: 100, 200 klienti

Klientu izvietojuma variācijas:

- Vienkāršs izlīdzinājums: Klienti izvietoti nejauši visā teritorijā.
- Trīs pilsētu klasteri: Klienti koncentrēti trīs dažādos klasteros.
- Septiņas pilsētu klasteri: Klienti koncentrēti septiņos dažādos klasteros.

Testēšanas procesā mērītas šādas metrikas:

- Izpildes laiks: Kopējais laiks, kas tika patērēts, lai atrastu optimālu risinājumu.
- Risinājuma kvalitāte: Salīdzināms sākotnējā un optimizētā risinājuma kopējais attālums, lai noteiktu izmaksu samazinājumu.

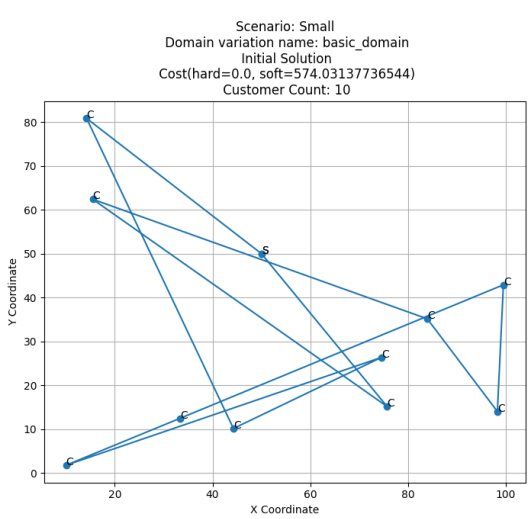
Testu veikšana

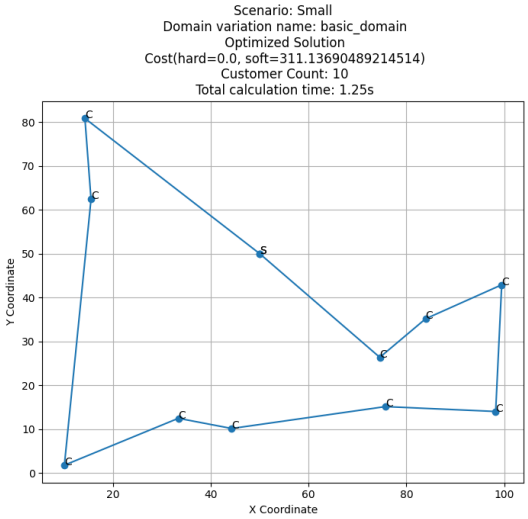
Katram testa gadījumam algoritms tika palaists, un rezultāti tika saglabāti atsevišķos failos. Tāpat tika ģenerēti grafiki, kas vizualizē sākotnējo un optimizēto risinājumu, kā arī izpildes laiku un izmaksu samazinājumu attiecībā pret klientu skaitu.

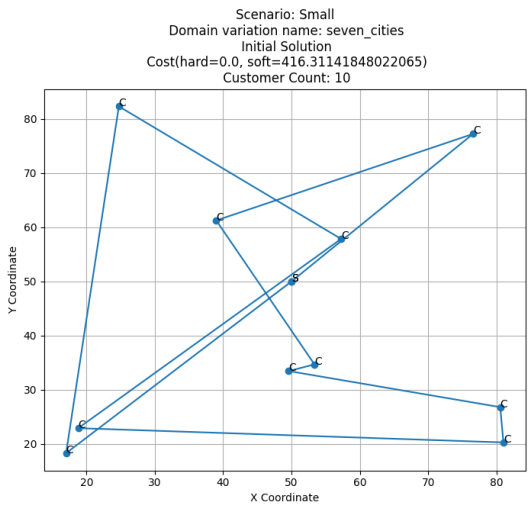
Autors: Toms Kārlis Bicāns
St.apl.nr: tb19019

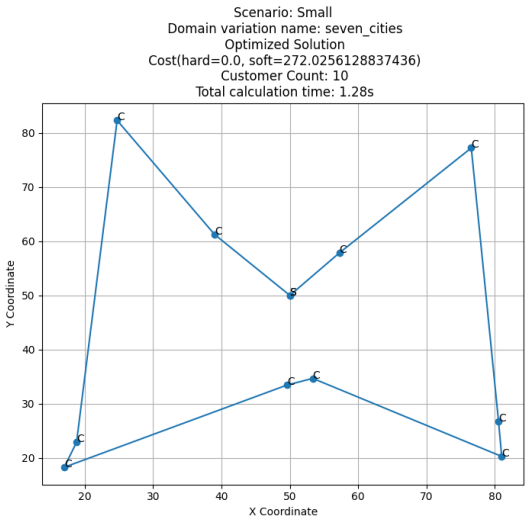
Pāris no iegūtajiem testa rezultātiem (Var arī palaist programmu, lai grafikus apskatītos interaktīvi – tad ir jānomaina programmas augšā konstante `SHOW_CHARTS` uz `true`):

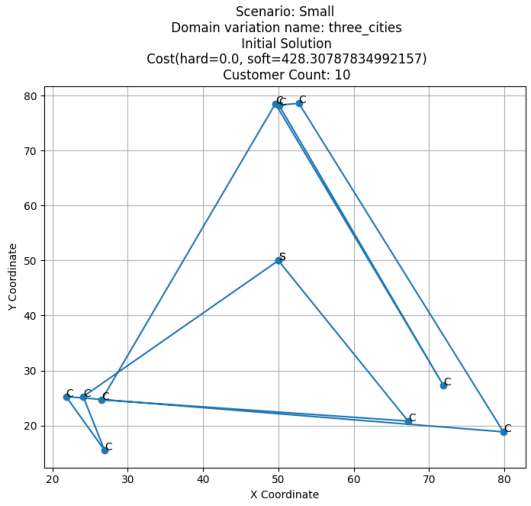
10 klienti:

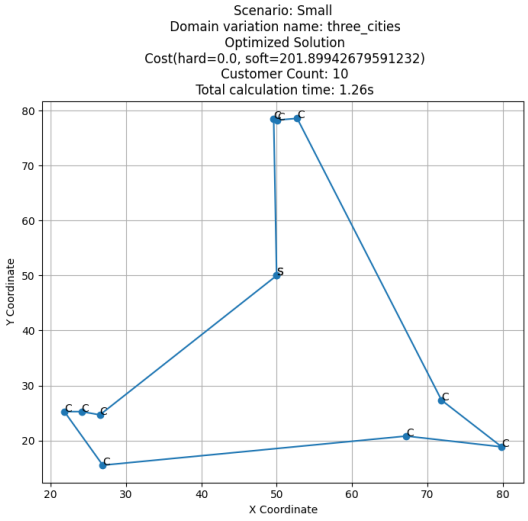






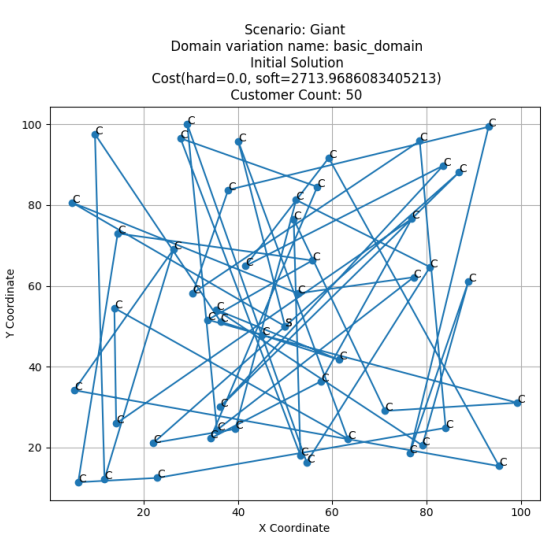


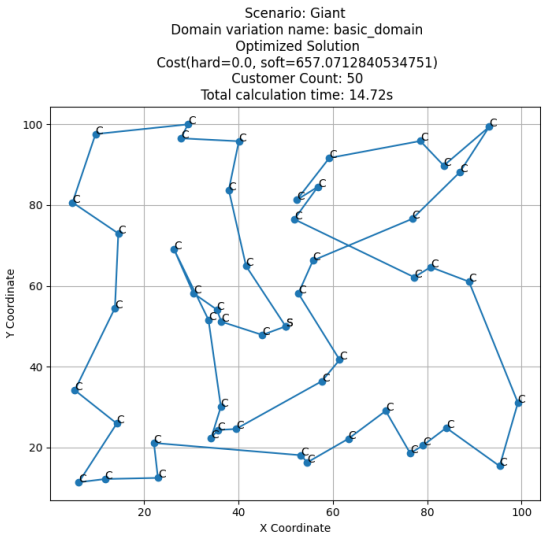


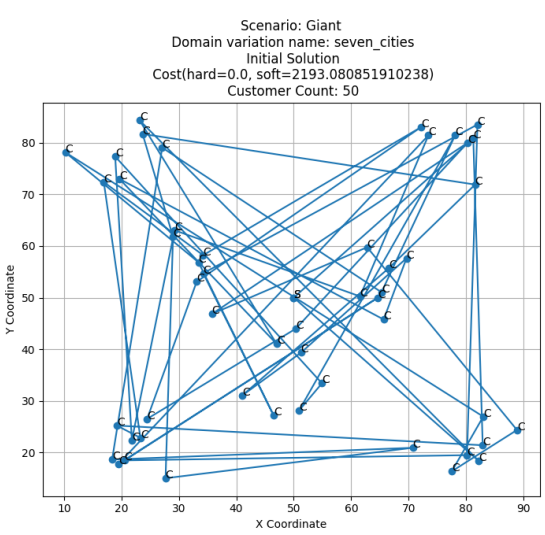


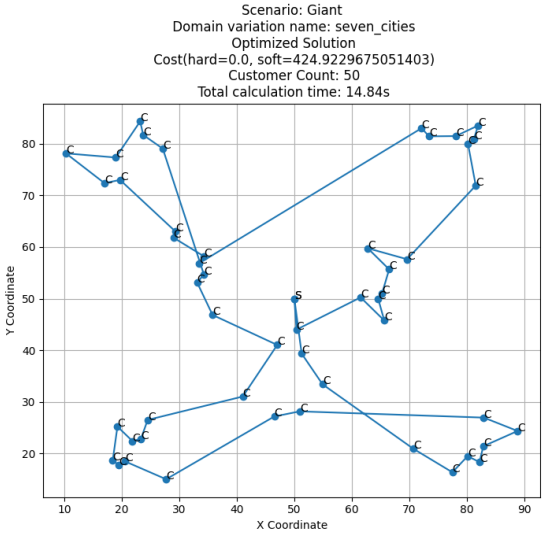
50 klienti:

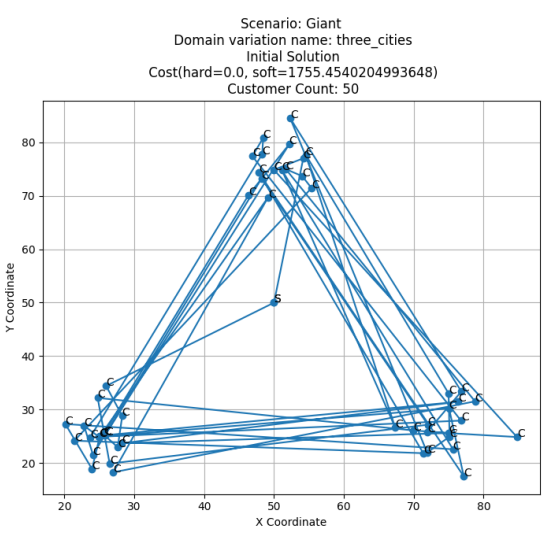
Autors: Toms Kārlis Bicāns
St.apl.nr: tb19019

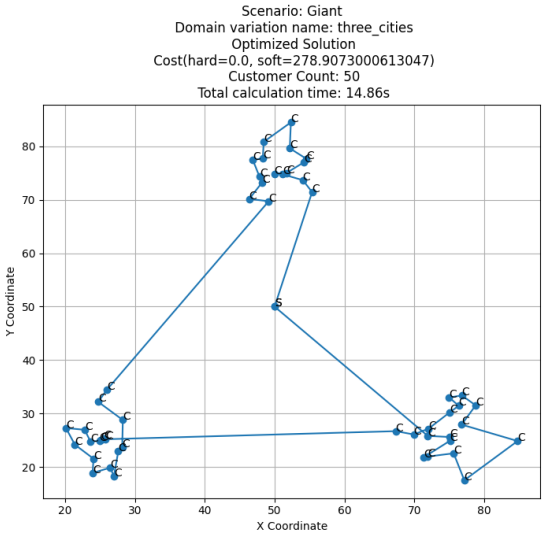






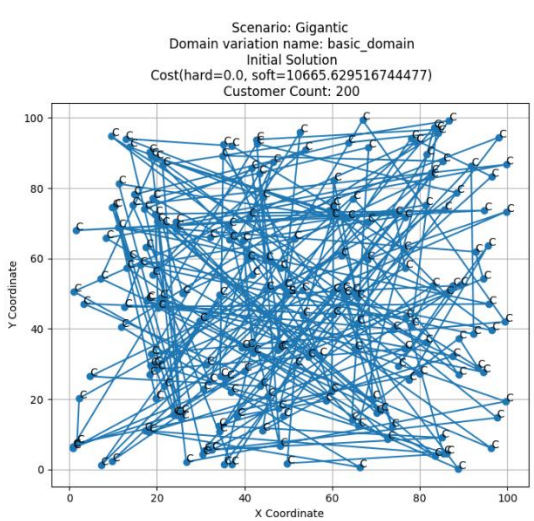


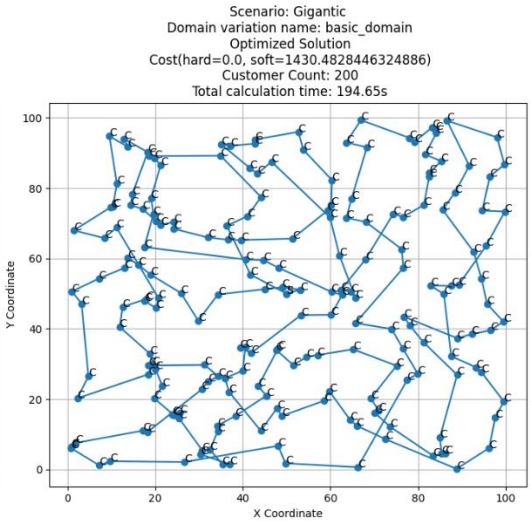


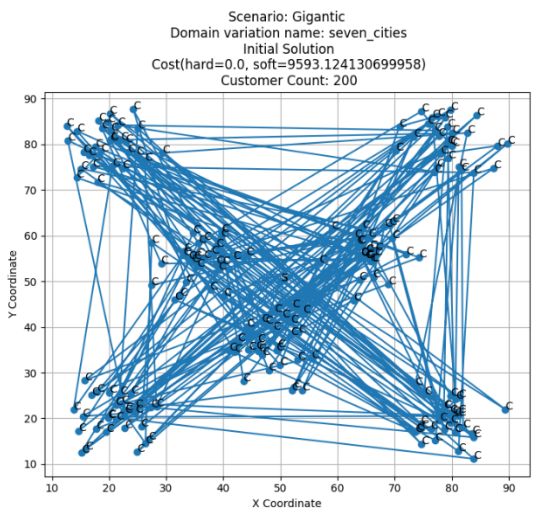


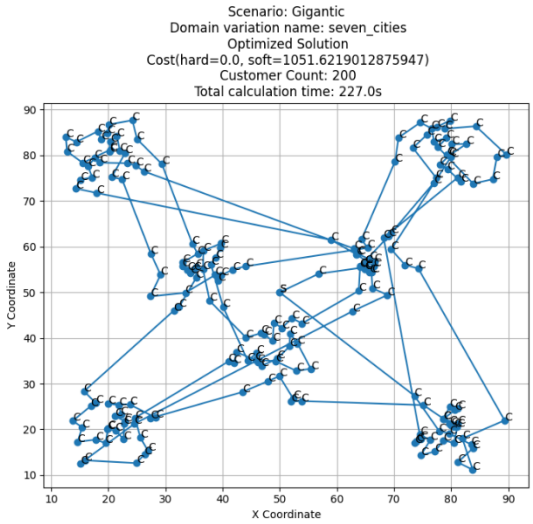
Autors: Toms Kārlis Bicāns
St.apl.nr: tb19019

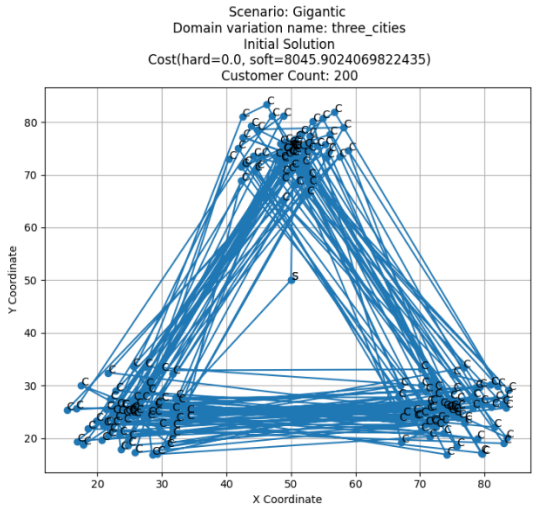
200 klienti:

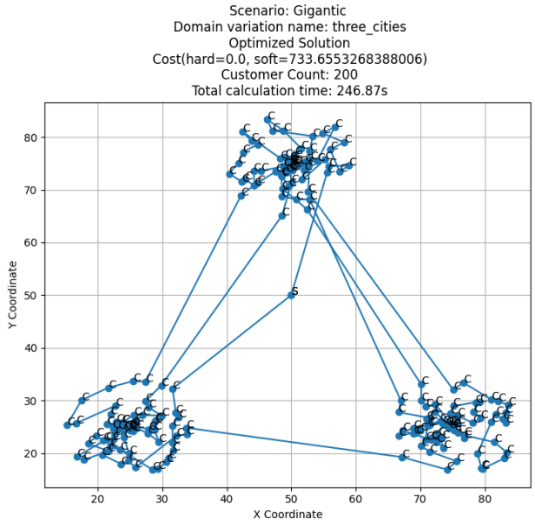






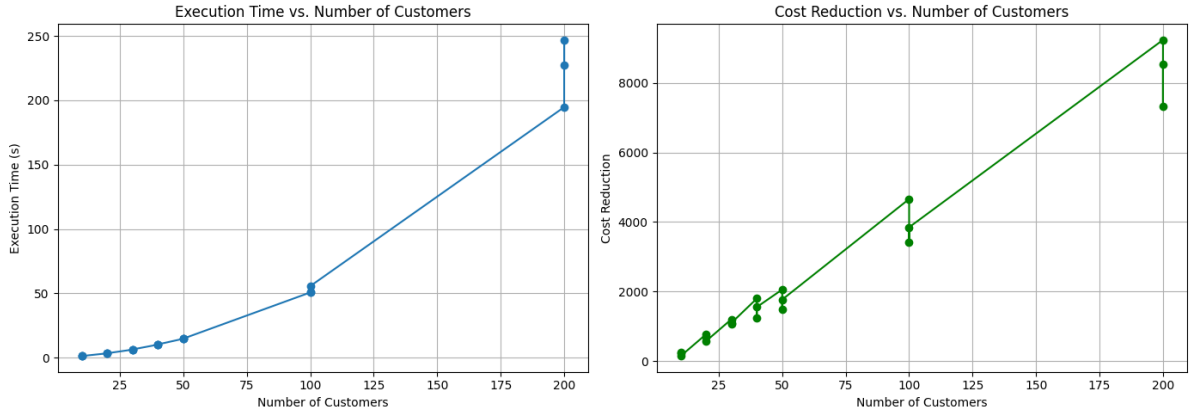






Autors: Toms Kārlis Bicāns
St.apl.nr: tb19019

Apkopojums:



Secinājumi:

Pie maziem problēmu izmēriem, izstrādātais algoritms darbojas ļoti ātri (10 klientiem maršruts tika izrēķināts ~1.25 sekundēs un uz aci izskatās, ka tiek atrasts visoptimālākais maršruts), taču pie lielākiem problēmu izmēriem bija nepieciešams ilgāks meklēšanas laiks (200 klientu maršrutu rēķināja ~235 sekundes), kā arī pilsētu gadījumā transports šķietami liekas reizes pārvietojās no pilsētas uz pilsētu. Šo liekās pārvietošanās problēmu netika paspēts atrisināt.

Kopumā risinājums šķiet strādā bez problēmām, ja klientu skaits ir zem 50. Lielākam klientu skaitam būtu nepieciešams veikt uzlabojumus apkārtnes funkcijā.

Vietas, kur algoritmu var uzlabot un padarīt interesantāku:

1. Ieviest vairākas transportu stacijas – šajā risinājumā tika izmantota tikai 1 stacija. To var izdarīt papildinot domēnu ar sarakstu ar stacijām un pielāgot apkārtnes funkciju.
2. Ieviest vairākus transportlīdzekļus – tika veidots tikai 1 maršruts, taču risinājumu var uzlabot ieviešot vairākus maršrutus. Būtu jāpapildina apkārtnes funkcija (domēns jau ietver sarakstu ar maršrutiem risinājuma klasē)

Autors: Toms Kārlis Bicāns
St.apl.nr: tb19019

4. Saite uz github repozitoriju:

https://github.com/TomsBicans/LU-DF-MAG-Praktiska-kombinatoriala-optimizacija