Projektēšanas pārskats

Kursā “Projektēšanas laboratorija”

“**Preču piegādes maršrutēšana**”

Grupa: Edvards Muraško,

Mārtiņš Vīksna,

Sandijs Krūmiņš,

Miks Matīss Liepiņš

RĪGA 2025

**Ievads**

**Problēmas nostādne**: Mūsdienu piegādes ķēdēs un loģistikā liela nozīme ir efektīvai maršrutu plānošanai, lai optimizētu resursu izmantošanu un savlaicīgi izpildītu klientu pasūtījumus. Tas kļūst īpaši sarežģīti, ja piegādes ir jāveic noteiktos laika logu ierobežojumos. Uzņēmumiem ar vairākiem piegādes punktiem, ierobežotu transportlīdzekļu skaitu un precīzām piegādes prasībām ir nepieciešami automatizēti risinājumi, lai atrastu optimālus maršrutus.

**Darba mērķis**: Šī darba mērķis ir izstrādāt risinājumu maršrutēšanas problēmai ar laika logiem (Vehicle Routing Problem with Time Windows, VRPTW), izmantojot Solomon VRP algoritmu un Google OR-Tools. Optimizēt piegādes maršrutus tā, lai minimizētu kopējo piegādes laiku. Nodrošināt, ka piegādes notiek klientu noteiktajos laika logu ierobežojumos. Efektīvi izmantot pieejamos transportlīdzekļus un piegādes resursus.

**Novērtēšanas mērķis:** Izvērtēt maršrutēšanas programmas efektivitāti piegādes maršrutu plānošanā, ņemot vērā dažādu piegādes punktu skaitu (N). Analīze tiek veikta, lai noteiktu izmantoto kurjeru skaitu, kopējo maršruta laiku un individuālo vidējo maršruta laiku uz vienu kurjeru.

**Līdzīgo risinājumu pārskats**

Preču piegādes maršrutu plānošanai ir pieejami dažādi rīki, kuri piedāvā dažādas funkcijas, lai radītu efektīvus un optimizētus maršrutus priekš lietotājiem un uzņēmumiem. Daži no populārākajiem rīkiem ir Google Maps API, RouteXL, OptimoRoute, Onfleet un Waze.

Google Maps API piedāvā maršrutu plānošanas iespējas, ieskaitot dažādu punktu secību maršrutā, attālumu matricu aprēķināšanu un precīzu maršrutu kartēšanu. Tā spēj piekļūt reāllaika datiem, piemēram, satiksmes apstākļiem, kas palīdz optimizēt maršrutus, lai izvairītos no sastrēgumiem. Tā arī ir plaši pieejama. Google Maps API var izmantot varteikt jebkurā mobilajā telefonā vai datorā. Google Maps API izmanto divus grafu algoritmus: Deikstras algoritmu un A\* meklēšanas algoritmu.

RouteXL ir maršrutu plānošanas rīks, kas spēj optimizēt maršrutus ar vairākām pieturām. Lietotājam ir tikai jāievada adreses, kurās vajag apstāties, un rīks izveidos optimizētu maršrutu. Tas ir ļoti noderīgs mazākiem uzņēmumiem, jo ar to var apstrādāt maršrutus līdz 20 apstāšanās punktiem par brīvu. Lai izveidoto maršrutus ar vairāk apstāšanās punktiem, ir nepieciešams iegādāties maksas versiju. RouteXL izmanto algoritmu, kas izveido optimizētu maršrutu, ar kuru lietotājs pavadīs vismazāk laika ceļā.

OptimoRoute ir piegādes maršrutu plānošanas rīks ar plašu funkcionalitāti. Ar to var optimizēt vairāku transportlīdzekļu maršrutus, pārvaldīt kurjerus un veikt attālumu matricu aprēķinus. Tajā var ievadīt apmeklējamo klientu sarakstu, transportlīdzekļa jaudu un kapacitāti, un laikus, cikos ir jāpiegādā klientiem produkti. OptimoRoute izmanto maršrutēšanas funkcijas, kas ņem vērā ceļa ierobežojumus, laiku, cikos ir jāpiegādā produkti, kā arī transportlīdzekļa kravas veidu, paaugstinātai drošībai. Rīks ir ļoti noderīgs vidējiem un lieliem uzņēmumiem, kuriem ir nepieciešama sarežģīta maršrutu optimizācija.

Onfleet ir piegādes un pārvaldības programmatūra, kas nodrošina loģistikas un maršrutu optimizācijas pakalpojumus. Tā ļauj uzņēmumiem ievadīt apmeklējamo klientu sarakstu, piešķirt kurjerus, optimizēt maršrutus un pārvaldīt piegādes procesus. Tā izmanto algoritmu, kas spēj optimizēt maršrutus, ņemot vēra vairākus transportlīdzekļus, ceļu un kurjeru noslogojumus, un piegādes prioritātes. Tā ir piemērota lieliem uzņēmumiem, jo programmatūra piedāvā daudz funkciju, ar kurām var optimizēt gan vieglus, gan arī sarežģītus maršrutus.

Waze ir bezmaksas lietotne, un to var izmantot bez reģistrācijas, taču reģistrējoties ir vairāk iespēju izmantot tās sociālā tīkla funkcijas. Galvenā lietotnes funkcija ir navigācija, kas parāda satiksmes intensitāti, sastrēgumus un cita veida traucējumus, piedāvā mainīt doto maršrutu, ja ir labākas opcijas. Navigācija arī ļauj pirms brauciena izvēlēties no vairākām maršruta opcijām. Būtiska funkcija ir lietotāju iespēja ziņot par šiem satiksmes traucējumiem, ieskaitot ceļu policijas posteņus, kā arī sazināties ar citiem Waze lietotājiem. Kartes un ģeotelpiskos datus, ko izmanto lietotne, daļēji veido un rediģē paši tās lietotāji. Waze, Tāpat kā Google Maps, izmanto Deikstras algoritmu un A\* meklēšanas algoritmu.

Attēls, kurā ir teksts, ekrānuzņēmums, programmatūra, fonts

Apraksts ģenerēts automātiski

**Tehniskais risinājums**

**Prasības**

1. Piekļuve tikai autorizētiem lietotājiem
2. Maršrutu izveide izmantojot CRUD operācijas, sarakstu pārlūkošanas izveide.
3. Maršruta datu ievade(apmeklējamo klientu skaits)
4. Maršrutēšanas parametra ievade atbilstoši modelim(kurjeru skaits un attālumu matrica)
5. Maršruta aprēķināšana
6. Maršrutu attēlošanas karte, grafs un efektivitātes rādītāji.

**Algoritms**

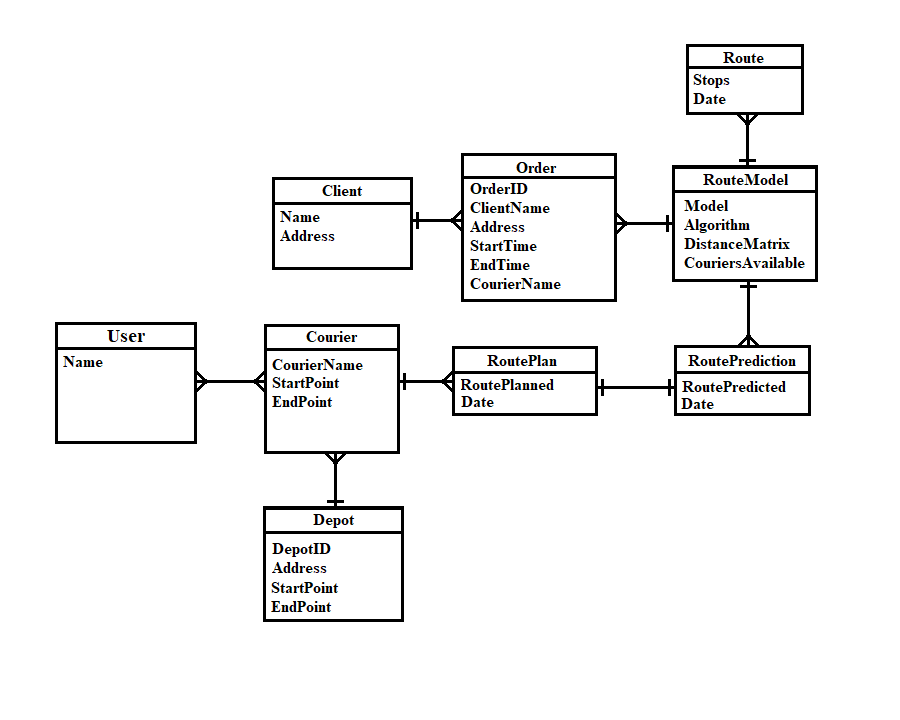
Šis algoritms implementē Transportlīdzekļa Maršrutēšanas Problēmu (Vehicle Routing Problem, VRP), izmantojot Google OR-Tools bibliotēku. Tas risina Solomon VRP variantu, kas ietver laika logu ierobežojumus, lai optimizētu vairāku transportlīdzekļu maršrutus. Algoritms minimizē kopējās piegādes izmaksas vai laiku starp depo un klientiem, ievērojot transportlīdzekļu ietilpības un klientu piegādes laika logu ierobežojumus.

Algoritms sāk ar piegādes datu sagatavošanu, ietverot klientu lokācijas, laika logus un depo informāciju. Tiek izveidotas attāluma un laika matricu tabulas, lai modelētu ceļojumu laiku starp vietām. Optimizācijas procesā tiek izmantots OR-Tools modelis ar dažādām stratēģijām, piemēram, “Parallel Cheapest Insertion” un “Path Cheapest Arc”, lai atrastu optimālus risinājumus. Laika logu ierobežojumi tiek nodrošināti, un tiek piešķirti maršruti katram transportlīdzeklim.

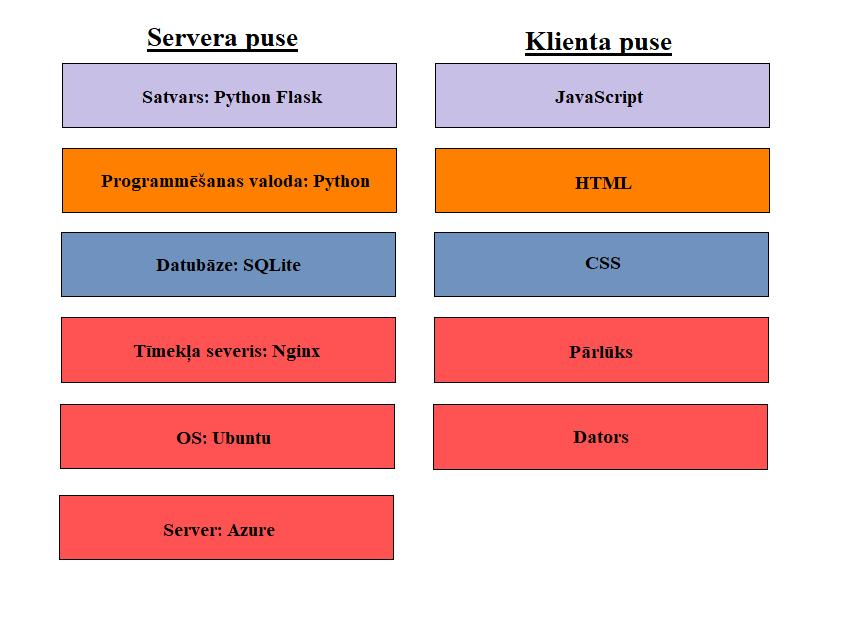
Ja risinājums tiek atrasts, algoritms izvada maršrutus, piegādes kārtību un plānotos piegādes laikus, kā arī optimizācijas statistiku. Šis algoritms ir piemērots lietojumiem, kur efektīva loģistika ir kritiska, piemēram, e-komercijas piegādēm vai kurjeru pakalpojumiem.

*Algoritma pseidokodu skat. 1. pielikumā.*

**Konceptu modelis**



**Tehnoloģiju steks**



**Programmatūras apraksts**

Šī programmatūra ir tīmekļa lietotne, kas paredzēta loģistikas maršrutu plānošanai un optimizācijai, palīdzot uzņēmumiem efektīvi pārvaldīt piegādes procesus. Lietotne nodrošina lietotāju reģistrācijas un autentifikācijas iespējas, izmantojot Flask-Login, ļaujot katram lietotājam droši pārvaldīt savus datus. Tā piedāvā funkcionalitāti piegāžu datu augšupielādei, kur CSV failos ietvertā informācija par klientiem, adresēm un laika logiem tiek ģeokodēta ar geopy bibliotēkas palīdzību, lai noteiktu ģeogrāfiskās koordinātas.

Programmatūras galvenais izcilības punkts ir maršrutu optimizācija, kurā tiek izmantota OR-Tools bibliotēka un SolomonVRP klase. Tā aprēķina efektīvus piegāžu maršrutus, ņemot vērā depo atrašanās vietu, transportlīdzekļu skaitu un piegādes laika logus. Optimizācijas rezultātā tiek ģenerēti detalizēti maršruti katram transportlīdzeklim, nodrošinot piegāžu secību un plānoto ierašanās laiku.

Šie maršruti tiek saglabāti datubāzē, izmantojot SQLAlchemy, un ir pieejami lietotājiem apskatei un rediģēšanai lietotnes interfeisā.

Lietotne piedāvā arī plašas rediģēšanas iespējas, tostarp piegāžu pievienošanu, atjaunināšanu un dzēšanu. Visa lietotāja augšupielādētā informācija tiek validēta un saglabāta datubāzē ar Flask-Migrate palīdzību, nodrošinot datu drošību un vieglu atjaunošanu. Maršrutu vizualizācija ļauj lietotājiem apskatīt detalizētas piegāžu detaļas, piemēram, piegādes secību, klientu informāciju, adreses un laika logus.

Šī programmatūra ir lieliski piemērota loģistikas uzņēmumiem, kas vēlas optimizēt piegādes procesus, samazināt izmaksas un uzlabot klientu apmierinātību, ievērojot to laika prasības. Tā apvieno jaudīgas tehnoloģijas, piemēram, Flask tīmekļa ietvaru, Google OR-Tools optimizācijas algoritmus un ģeogrāfisko datu apstrādi, radot visaptverošu rīku efektīvai loģistikas pārvaldībai.

**Novērtējums**

**Novērtējuma plāns**

**Ieejas mainīgie**: Maršrutu datu kopas lielums(30, 60, 100 ieraksti)(N)

Testa kopas veidus šim algoritmam nav ko pielikt pie mainīgajiem, jo priekš novērtēšanas mēriem informācija jau ir dota, apstrādes laiks ir konstants 90 sekundes, kurās algoritms rēķina labākos maršrutus, šāds laiks ir uzlikts, lai nepārslogotu programmatūru, un lai nav tā, ka viņš aizņem ļoti daudz laika rēķinot maršrutus. Kā arī atpazīšanas ātrums csv failiem par katru ierakstu failā ir 0,5 sekundes.

**Novērtēšanas mēri**: Izmantoto kurjeru skaits, Kopējais maršruta laiks(stundās un minūtēs), Individuālais vidējais maršruta laiks kurjeram(stundās un minūtēs).

**Novērtējuma rezultāti**

|  |  |  |  |
| --- | --- | --- | --- |
| **N** | **30** | **60** | **100** |
| **Izmantot kurjeru skaits** | **6** | **10** | **12** |
| **Kopējais maršruta laiks** | **2887min**  **Jeb**  **48h un 7min** | **4716min**  **Jeb**  **78h un 36min** | **5754min**  **Jeb**  **95h un 54min** |
| **Individuālais vidējais maršruta laiks(kurjera)** | **481min**  **Jeb**  **8h un 1min** | **471min**  **Jeb**  **7h un 51min** | **479,5min**  **Jeb**  **7h un 59min** |

**Secinājumi**

Izstrādātā maršrutēšanas programma demonstrē augstu efektivitāti, nodrošinot elastīgu resursu pārvaldību piegādes ķēdēs. Analīze parāda, ka izmantoto kurjeru skaits tiek dinamiski pielāgots piegādes punktu skaitam, kas ļauj samazināt nevajadzīgas izmaksas un nodrošināt optimālu transportlīdzekļu izmantošanu. Tas ir īpaši svarīgi uzņēmumiem, kas strādā ar lieliem piegādes apjomiem un sarežģītiem maršrutiem.

Algoritms spēj ievērojami optimizēt kopējo maršruta laiku, neskatoties uz piegādes punktu pieaugumu. Lai gan piegādes punktu skaita palielināšanās loģiski pagarina kopējo maršruta laiku, programma nodrošina, ka laika pieaugums ir kontrolēts. Tas norāda uz izstrādātā risinājuma piemērotību, lai apstrādātu lielus piegādes apjomus, nezaudējot efektivitāti.

Individuālā kurjeru slodze tiek vienmērīgi sadalīta neatkarīgi no piegādes punktu apjoma. Vidējais darba laiks uz vienu kurjeru saglabājas gandrīz nemainīgs, kas palīdz uzturēt vienmērīgu darbinieku noslodzi un novērst pārslodzes risku.

Programma arī nodrošina precīzu un ātru maršrutu aprēķinu, apstrādājot sarežģītus piegādes maršrutus ar laika logu ierobežojumiem. Algoritms ģenerē optimizētus risinājumus īsā laikā (90 sekundēs). Salīdzinot ar līdzīgiem rīkiem, programma piedāvā specializētu pieeju, izmantojot Solomon VRP un Google OR-Tools algoritmus, kas ir īpaši piemēroti loģistikas problēmām ar laika logiem.

**Pielikums**

1. **Pielikums**

Class SolomonVRP:

INPUT:

deliveries\_df: tabula ar kolonnām [package\_id, latitude, longitude, time\_from, time\_to]

num\_vehicles: pieejamo transportlīdzekļu skaits

depot\_location: tuple (latitude, longitude) depo koordinātēm

route\_name: maršruta nosaukums

FUNCTION \_\_init\_\_(self, deliveries\_df, num\_vehicles, depot\_location, route\_name):

INITIALIZE:

deliveries ← deliveries\_df

num\_vehicles ← num\_vehicles

depot\_lat, depot\_lon ← depot\_location

route\_name ← route\_name

locations ← \_prepare\_locations()

time\_matrix ← \_create\_time\_matrix()

time\_windows ← \_prepare\_time\_windows()

INITIALIZE log

FUNCTION \_prepare\_locations():

DEPOT kā pirmais koordinātu punkts

PIEVIENO piegādes punktus no tabulas

RETURN visas koordinātas

FUNCTION \_haversine\_distance(lat1, lon1, lat2, lon2):

R ← 6371 (Zemes rādiuss kilometros)

KONVERTĒ koordinātas radiānos

Aprēķini lielā loku attālumu starp punktiem

KONVERTĒ attālumu minūtēs, pievieno buferi

RETURN attālums minūtēs

FUNCTION \_create\_time\_matrix():

IZVEIDO tukšu matricas sarakstu

FOREACH i, j pāris no koordinātām:

IF i ≠ j:

Aprēķini attālumu starp i un j

RETURN matricu

FUNCTION \_convert\_time\_to\_minutes(time\_str):

MĒĢINI konvertēt laiku uz minūtēm pēc dažādiem formātiem

IF formatēšana neizdodas:

Paziņo par kļūdu un izmanto noklusējumu

RETURN laiku minūtēs

FUNCTION \_prepare\_time\_windows():

DEPOT darba laiks ir pilna diena

FOREACH piegādes punkts:

KONVERTĒ sākuma un beigu laikus

JA beigu laiks ≤ sākuma laiks:

Koriģē beigu laiku

Pievieno logu sarakstam

RETURN laiku logi

FUNCTION \_log\_model\_stats(manager, routing, solution):

IZVADA galveno informāciju par modeli un risinājumu

IF nav risinājuma:

IZVADA laika logus

ELSE:

FOREACH transportlīdzekļa ceļš:

IZVADA ceļa statistiku

IZVADA kopējo laiku

FUNCTION solve():

IZVEIDO modeli

Pievieno laika aprēķina funkciju

IZVEIDO laika logu dimensiju

Pievieno laika logus katram punktam

DEFINĒ meklēšanas stratēģijas

FOREACH stratēģija:

MĒĢINI atrisināt modeli

IF risinājums atrasts:

IZVADI ziņu un izbeidz meklēšanu

IF nav risinājuma:

IZVADI kļūdas ziņu

RETURN null

IZVĒLIET risinājumu

FOREACH transportlīdzeklis:

IZVADI ceļa punktus un plānoto laiku

RETURN transportlīdzekļu maršruti

FUNCTION validate\_input\_data():

PĀRBAUDI transportlīdzekļu skaitu

PĀRBAUDI piegādes punktu koordinātas

PĀRBAUDI laika logus katrai piegādei

IZVADI brīdinājumus

RETURN vai dati ir derīgi