

UNIVERSITETI I PRISHTINËS “HASAN PRISHTINA”
FAKULTETI I SHKENCAVE MATEMATIKO-NATYRORE
DEPARTAMENTI I MATEMATIKËS
PROGRAMI: SHKENCA KOMPJUTERIKE



Lënda: Inteligjenca Atrificiale
Automatizimi i dritave të trafikut duke përdorur algoritëm gjenetik

Punuar nga:

1. Rinesë Morina
2. Rona Latifaj
3. Eljesa Kqiku

Mars 2023

1 Përshkrimi i problemit

Nëse jetoni në një qytet të madh, mund të supozojmë me shumë siguri një ndër problemet më të zakonshme për ju është trafiku. Kur udhëtoni në drejtim të qytetit, shpesh mund të ndodhë që vonesat gjatë rrugës të zgjasin më gjatë se puna juaj në destinacion. Madje ky trafiku përveç të qenurit zhgënjyes, ka pasoja edhe në mjedis. Qartazi të gjitha ato automjete në pritje kanë ndikim në cilësinë e ajrit.

Kur jeni të bllokuar dhe ulur pas një vargu të gjatë makinash, është e lehtë ta lini mendjen tuaj të endet mbi zgjidhjet për problemet tona të trafikut. Por, menaxhimi i trafikut në zonat e dendura urbane është një problem jashtëzakonisht kompleks me synime dhe sfidave kontradiktore.

Një nga ato sfida më themelore ndodh në një kryqëzim, ku shumëfishohen rrymat e trafikut - duke përfshirë automjetet, biçikletat dhe këmbësorët - dhe secila duhet të jetë e sigurt.

Me kalimin e viteve, sfidën se kush do të presë dhe kush vazhdon rrugën, njerëzit e kanë menaxhuar në shume mënyra, nga tabelat e thjeshta të përparësisë, te rrethrotullimet, si edhe, një nga më të zakonshmet, me sinjalet e trafikut (semaforët).

Në këtë detyrë, ne po përpiqemi të studiojmë se si mund të përdorim ndonjë nga algoritmet lokale të kërkimit, në veçanti algoritmin gjenetik, për të modeluar dhe gjetur një zgjidhje për sekuencën e radhitjeve, kohëzgjatjes sa do të jetë i hapur dhe sa i mbyllur secili nga semaforët, duke e shikuar problemin si Constraint Satisfaction Problem.

Për modelin që po e prezantojmë në vijim, po ofrojmë një lloj zgjidhje të përgjithësuar, duke lejuar që gjendja e trafikut të merret si input, ashtu që algoritmi të mund të përdoret në situata të ndryshme dhe jo në një udhëkryq të vetëm. Modifikimet e lejuara do të përfshijnë kohëzgjatjen që semaforët mund të jene të mbyllur dhe kohën e preferueshme të qëndrojnë të hapur, ngarkesën e trafikut, rrugët që po lejojmë të hapen në kohën e njëjtë, dhe ato që nuk preferojmë por mund të hapen njëkohësisht për të zbutur kushtet. Poashtu po ofrojmë një koeficient prioriteti, i cili do të lejojë që algortimi të përshtatet me situata urgjente, siç do të ishte për shembull prania e një auto-ambulance ose makine policie, por edhe kalimi i një grupi të madh këmbësorësh në ndonjë event.

Pas modelimit, po përdorim algoritmin gjenetik, për të provuar të gjejmë një zgjidhje të problemit për një kohë të caktuar gjatë ditës, programi do të ekzekutohet me parametrat e ngarkesës së trafiku për të gjetur një zgjidhje e cila mund të përdoret për një interval disa orësh, para se të masim parametrat e rinjë dhe të ri-ekzekutojmë programin, ndërsa "popullata" që algoritmi ka gjeneruar në secilin hap, mund të përdoret si popullatë fillestare në ekzekutimin e radhës.

2 Modeli

Për të modeluar problemin tonë po e analizojmë situatën nga dy elemente kyçe: e para, nga bashkësia e semaforëve të vendosur në udhëkryq; dhe e dyta, nga sekuenca



Figure 1: Diagrami unazë-pengesë

e intervaleve kohore kur ata janë të hapur apo jo.

Inxhinierët e trafikut përdorin një diagram unazë-pengesë për të shikuar se si faza të ndryshme të sinjali lejohen të funksionojë. Skicat e tilla kërkojnë që dizajnerët të krijojnë disa grupime dhe secili prej tyre të ketë kohë të barabarta para një pengese kur i gjithë trafiku vendoset në pritje, për të vijuar me një sekuenca tjetër.

Ne po përdorim diçka të ngjashme, por në vend se t'i grupojmë rrugët që lejohet të lëvizin në të njëjtë kohë, vetëm po e njoftojmë algoritmin se cilat rrugë lejohen apo preferohen të hapen në kohën e njëjtë në formë të një matrice të shkallëzuar apo matricë të plotë por vetëm të jetë simetrike, një validim të tillë po e kërkojmë në mënyrë që mos t'i prishim kushtëzimet indirekt, prandaj në implementimin në kod gjatë leximit të matricës do të vendosim kushtin më të keq nëse përdoruesi i ka vendosur të ndryshme. Një shembull i tabelës së tillë jepet me tabelën 1.

Table 1: Matrica e sekuençave

	Semafori 1	Semafori 2	Semafori 3
Semafori 1	ALWAYS	ACCEPTABLE	NEVER
Semafori 2	ACCEPTABLE	ALWAYS	UNPREFERRED
Semafori 3	NEVER	UNPREFERRED	ALWAYS

Ndërsa thelbi i zgjidhjes sonë është një strukturë dy dimensionale që varet nga koha dhe secili semafor, këtë strukturë do t'a quajmë **matricë e sekuençave kohore**. Secila shtyllë e matricës përfaqëson një interval uniform kohor, gjatësia e të cilit merret si njësi. Nëse njësi jonë do të ishte 10 sekonda, dhe dëshirojmë një semafor të jetë i hapur për 30 sekonda, atëherë do të duhet të merr pjesë në 3 sekuenca radhazi. Secili rresht i matricës përfaqëson një semafor, dhe prerja me secilin interval kohor tregon a është semafori specifik i hapur apo jo në atë interval.

Table 2: Matrica e sekuençave

	Time Frame 1	Time Frame 2	...	Time Frame n
Semafori 1	0	1		0
Semafori 2	0	1		0
⋮			⋱	
Semafori k	0	1		0

Një sekuençë do të jetë zgjidhje e pranueshme nëse në asnjërin nga intervalet nuk hapen dy semaforë të shenuar si NEVER përnjëherësh. Ndërsa zgjidhja ideale nuk do të kishte as prerje të UNPREFERRED, dhe do të ketë numrin maksimal të semaforëve të hapur njëkohësisht. Por do të kemi kujdes që nëse njësia e intervalit kohor është e vogël, semafori të jetë i hapur në disa sekuenca radhazi ashtuqë t'i japë pak kohë trafikut të zbrazet, prandaj edhe do të përdorim kushtëzimet e kohës minimale që semafori është i hapur, dhe kohës minimale të preferueshme, në të njëjtë kohe edhe maskimumin e kohës që do të jetë i mbyllur, si dhe maksimumin e preferueshëm.

Funksionet e evaluimit të sekuençave do t'a ndajmë ashtuqë semafori të njehësojë denimet dhe bonuset nga kohët minimale e maksimale, si dhe kushtëzimet me cilët semaforë tjerë është i hapur në të njëjtën kohë. Shuma e këtyre dy funksioneve jep pikët e një semafori në një time frame, ndërsa fitnesi i sekuençës së përgjithshme është shuma e secilit semafor në secilën time frame.

Denimet dhe bonuset që kemi vendosur në algoritëm paraqiten në dy tabelat e mëposhtme, ndërsa për shkeljet kohore:

- Sekuenca më e shkurtër se koha minimale për semaforin e hapur = - HARD
- Sekuenca më e shkurtër se koha minimale e preferuar por më e gjatë se koha minimale e lejuar për semaforin e hapur = - SOFT
- Sekuenca më e gjatë se koha maksimale për semaforin e mbyllur = - HARD
- Sekuenca më e gjatë se koha maksimale e preferuar por më e shkurtër se koha maksimale e lejuar për semaforin e hapur = - SOFT

Sa i madh do të jetë secili nga denimet e bonuset, mbetet poashtu input si konfigurim.

Table 3: Tabela e dënimit apo bonusit për gjendjet e semaforit të hapur

	ALWAYS	ACCEPTABLE	UNPREFERRED	NEVER
false	- HARD	- SOFT	+ SOFT	+ COOL
true	+ COOL	+ SOFT	- SOFT	- HARD

Table 4: Tabela e dënimit apo bonusit për gjendjet e semaforit të mbyllur

	ALWAYS	ACCEPTABLE	UNPREFERRED	NEVER
false	+ COOL	+ COOL	- COOL	- SOFT
true	- HARD	- SOFT	+ SOFT	+ SOFT

2.1 Klasa Population

Klasa population është objekt kryesorë në algoritmin gjenetik dhe komponentët e të dhënave të saja janë: popullata e shprehur si listë e kromosomeve, madhësia në numër e popullatës. Klasa ka një konstruktor i cili inicializohet me numrin e pritur të madhësisë së popullatës. Population ka gjithsej 7 metoda:

- initializePopulation e cila kërkon një sekuençë të semaforave dhe madhësinë e popullatës që dëshirojmë të krijojmë.
- createRandomChromosome përdoret për të krijuar individ dhe raste randome. Pra, zgjedhja e individeve për një popullatë është krejtësisht randome në krijimin e saj.
- selectIndividual zgjedh një individ të rastësishëm nga gjenerata aktuale.
- performCrossover merr një crossover rate, dhe bazuar në të kryen një ndër funksionet bazë në algoritmin gjenetik, duke krijuar kështu gjeneratën e re.
- performSelection merr si input gjeneratën e krijuar nga performCrossover dhe me koeficienta të kërkuar bën seleksionimin e prinderve të rinj duke u bazuar në funksionin getScore në një Priority Queue në të cilën kemi krijuar comparatorin tonë përkatës për objektet kromozom.
- performMutation duhe u bazuar në një random rate bën një ndryshim në kromozom duke krijuar mutacion.

2.2 Paketa Traffic

- Klasa "TrafficLight" është një klasë abstrakte që përfaqëson një semafor të trafikut. Ajo përmban variablat dhe metodat për të manipuluar me të dhënat e semaforit. Variablat e klasës janë: "isGreen": tregon nëse semafori është i hapur për kalim. "timeFrame": gjatësia e një frame në sekonda. "priority": prioriteti i semaforit. "trafficLightsConstraint": kufizimet e semaforëve të tjerë të lidhur me këtë semafor. "index": indeksi i semaforit në listën e semaforëve. "minTimeGreen": numri minimal i frame-ave që semafori duhet të qëndrojë i hapur pa marrë penallti të rënda. "maxTimeRed": numri maksimal i frame-ave që semafori guxon të qëndrojë i mbyllur pa marrë penallti të rënda. "prefMinTimeGreen": numrat e frame-ave të rekomanduar nga përdoruesi për të cilat semafori duhet të qëndrojë i hapur. "prefMaxTimeRed": numrat e frame-ave të rekomanduar nga përdoruesi për të cilat semafori duhet të qëndrojë i mbyllur.
- Konstruktori "TrafficLight" inicializon vlerat e variablave të klasës. Ka dy versione të konstruktorit, njëri merr parametrat e nevojshëm për inicializimin e semaforit, ndërsa tjetri përfshin edhe prioritetin e semaforit.
- Metoda "getScore" kthen një rezultat shifre që përfaqëson rezultatin e një sekuenca të dhënë të frame-ave për semaforin. Ky rezultat llogaritet duke marrë parasysh vlerat e kufizimeve kohore dhe kufizimet e kompatibilitetit të semaforit me semaforët e tjerë.
- Metoda "timeConstrainScore" llogarit rezultatin e kufizimeve kohore të semaforit në bazë të një sekuenca të dhënë të frame-ave. Rezultati mund të jetë pozitiv ose negativ, varet nga penalltitë që ka marrë semafori në këtë periudhë kohore.

- Metoda "compatibilityConstraintScore" llogarit rezultatin e kufizimeve të kompatibilitetit të semaforit me semaforët e tjerë në bazë të një sekuenca të dhënë të frame-ave. Rezultati varet nga tipi i kufizimit dhe konformiteti i sekuenca me kufizimet përkatëse.

Metodat e tjera janë getter-et për të marrë vlerat e variablave të klasës.

- Klasa "VehicleTrafficLight" është një nënklasë e klases "TrafficLight" dhe përfaqëson një semafor të trafikut për makinat. Ajo shton një variabël shtesë dhe disa metoda të riperdorura. Variablat e klasës janë: "trafficCoeff": koeficienti i ngarkesës së trafikut për semaforin e makinave. Konstruktori "VehicleTrafficLight" inicializon vlerat e variablave të klasës duke i kaluar parametrat e nevojshëm për konstruktorin e klasës bazë "TrafficLight". Ka dy versione të konstruktorit, njëri për inicializimin e semaforit pa prioritet dhe tjetri me prioritet.
- Metoda "getScore" kthen rezultatin e skores së semaforit të llogaritur nga metoda bazë "getScore" e klases "TrafficLight", por e shumëzon me koeficientin e trafikut për semaforin e makinave.
- Metoda "timeConstrainScore" është e riperdorur dhe llogarit rezultatin e kufizimeve kohore për semaforin e makinave. Rezultati llogaritet duke i kaluar parametrat e nevojshëm për metodën bazë "timeConstrainScore" dhe pastaj shumëzohet me koeficientin e trafikut.
- Klasa "VehicleTrafficLight" është krijuar për të përcaktuar specifikat e semaforëve të trafikut për makinat dhe për të shtuar funksionalitete të veçanta për to.
- Klasa "PedestrianTrafficLight" është një nënklasë e klases "TrafficLight" dhe përfaqëson një semafor të trafikut për këmbësorët. Ajo nuk shton variabla shtesë, por ka disa metoda të riperdorura.
- Konstruktori "PedestrianTrafficLight" inicializon vlerat e variablave të klasës duke i kaluar parametrat e nevojshëm për konstruktorin e klasës bazë "TrafficLight". Ka dy versione të konstruktorit, njëri për inicializimin e semaforit pa prioritet dhe tjetri me prioritet.
- Metoda "timeConstrainScore" është e riperdorur dhe llogarit rezultatin e kufizimeve kohore për semaforin e këmbësorëve. Rezultati llogaritet duke i kaluar parametrat e nevojshëm për metodën bazë "timeConstrainScore".
- Klasa "PedestrianTrafficLight" është krijuar për të përcaktuar specifikat e semaforëve të trafikut për këmbësorët dhe për të përdorur funksionalitetet e implementuara në klasën bazë "TrafficLight".
- Klasa "Sequence" paraqet një model të mundshëm të zgjidhjes për problemet e semaforëve të trafikut. Ajo implementon ndërfaqen "Node" dhe ka disa variabla

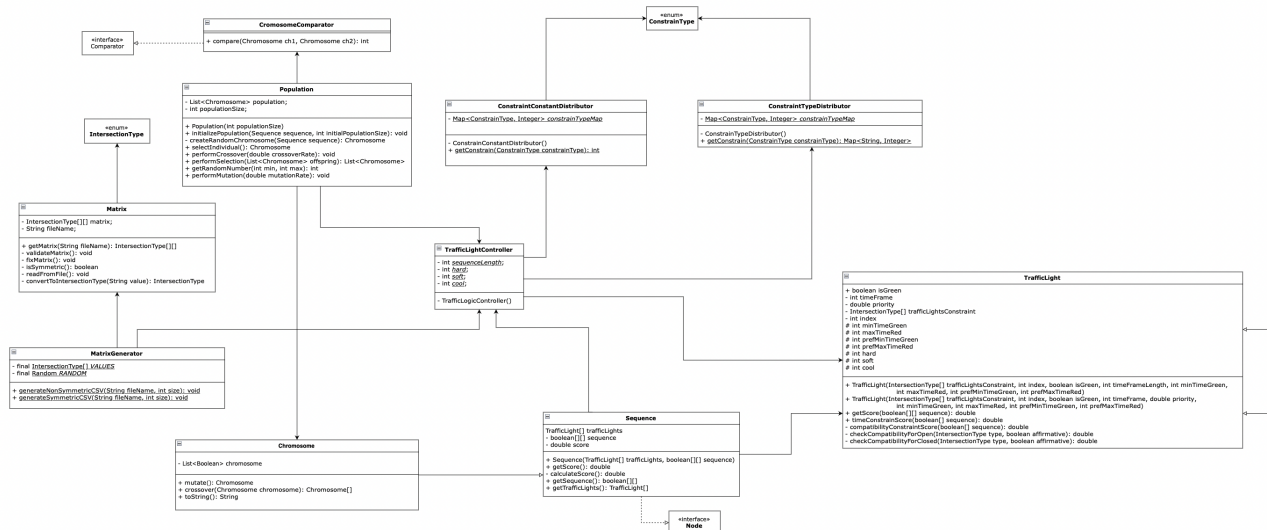
instance të deklaruara. "trafficLights" është një varg i objekteve "TrafficLight" që përfaqësojnë semaforët e trafikut në sekuençën e zgjidhjes. "sequence" është një matricë e vlerave booleane që paraqet gjendjen e secilit semafor në secilin interval kohor. "score" është vlera e fitnesit të zgjidhjes së paraqitur nga kjo sekuençë. Konstruktori "Sequence" inicializon vlerat e variablave instance duke marrë si parametra vargun e semaforeve dhe sekuençën e gjendjeve. Pasi inicializohen vlerat, llogaritet edhe vlera e fitnesit të zgjidhjes duke thirrur metodën "calculateScore".

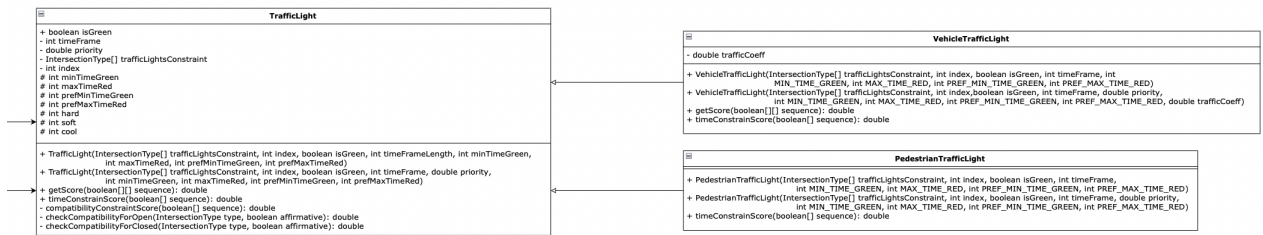
- Metoda "getScore" është e implementuar nga ndërfaqja "Node" dhe kthen vlerën e fitnesit të zgjidhjes.
- Metoda private "calculateScore" llogarit vlerën e fitnesit të zgjidhjes duke shënuar rezultatin e metodës "getScore" për secilin semafor në sekuençë dhe duke i shumëzuar rezultatet së bashku.
- Metodat "getSequence" dhe "getTrafficLights" kthejnë vlerat për variablat e instancës "sequence" dhe "trafficLights" respektivisht.
- Klasa "Sequence" është krijuar për të modeluar dhe operuar me zgjidhjet e mundshme për problemin e semaforeve të trafikut.

3 Arkitektura

3.1 UML

Diagrami i klasave ne UML mund te gjindet me poshte.





3.2 Paketa Matrix

Paketa "model.traffic.constraint.constraintMatrix" përmban klasën "Matrix" për matricën e kufizimeve të trafikut. Kjo klasë ofron funksione thelbësore për manipulimin e matricës së kufizimeve.

- Metoda "getMatrix" kthen matricën e kufizimeve nga një file.
- Metoda "validateMatrix" verifikon dhe rregullon matricën e kufizimeve.
- Metoda "fixMatrix" vendos vlerën "IntersectionType.ALWAYS" në diagonale dhe përdor vlerat simetrike në pjesën e poshtme të matricës.
- Metoda "isSymmetric" verifikon nëse matrica është simetrike.
- Metoda "readFromFile" lexon matricën e kufizimeve nga një file dhe shfaq mesazhe të gabimeve nëse ka probleme gjatë leximit.
- Metoda "convertToIntersectionType" konverton një string në tipin "IntersectionType".

Paketa përmban gjithashtu klasën:

- "MatrixGenerator" për gjenerimin e matricave të kufizimeve të trafikut në formën e file-ve CSV. Klasa ofron metoda për gjenerimin e matricave simetrike dhe asimetrike.
- Metoda "generateNonSymmetricCSV" gjeneron një file CSV me një matricë jo-simetrike të kufizimeve.
- Metoda "generateSymmetricCSV" gjeneron një file CSV me një matricë simetrike të kufizimeve.
- Klasa përdor klasën "IntersectionType" nga një paketë tjetër dhe klasat "FileWriter" dhe "IOException" nga Java API për shkrimin dhe kapjen e gabimeve gjatë shkrimit.

4 Algoritmi gjenetik

4.1 Pseudokodi

Algoritmi gjenetik në problemin e semaforëve

```
1:  $\lambda \leftarrow$  madhësia e popullatës
2:  $P \leftarrow \{\}$ 
3: for  $\lambda$  herë do
4:    $P \leftarrow P \cup \{\text{një individ i rastësishëm}\}$ 
5: end for
6:  $Best \leftarrow \square$ 
7: repeat
8:   for secilen Sekuencë  $P_i \in P$  do
9:     if  $Best = \square$  ose  $\text{Fitnesi}(P_i) > \text{Fitnesi}(Best)$  then
10:       $Best \leftarrow P_i$ 
11:    end if
12:  end for
13:   $Q \leftarrow \{\}$ 
14:  for  $\lambda/2$  herë do
15:    Prindi  $P_a \leftarrow \text{SelectWithReplacement}(P)$ 
16:    Prindi  $P_b \leftarrow \text{SelectWithReplacement}(P)$ 
17:    Fëmijët  $C_a, C_b \leftarrow \text{Crossover}(P_a, P_b)$ 
18:     $Q \leftarrow Q \cup \{\text{Mutate}(C_a), \text{Mutate}(C_b)\}$ 
19:  end for
20:   $P \leftarrow Q$ 
21: until  $Best$  është Sekuenca ideale ose kemi arritur limitin kohor
22: return  $Best$ 
```

▷ Zgjidhja

4.2 Crossover

Operatori crossover është analog me riprodhimin dhe kryqëzimin biologjik. Në këtë koncept zgjidhet më shumë se një prind dhe prodhohen një ose më shumë pasardhës duke përdorur materialin gjenetik të prindërve. Në rastin tonë kemi marrur dy prind, e duke vepruar mbi ta me metodën preformCrossover kemi fituar dy pasardhës të këtyre prindërve. Crossover zakonisht aplikohet në një algoritmin gjenetik me një probabilitet të lartë. E në rastin tonë kemi marru cossover rate të ndodhë në 90% të rasteve.

4.3 Mutation

Në një algoritëm gjenetik, mutacioni është një operacion gjenetik që fut material të ri gjenetik në një popullatë. Ne e përdorim këtë teknikë për të ruajtur diversitetin gjenetik nga një brez në tjetrin. Për më tepër, ne e përdorim atë për të parandaluar ngecjen e algoritmit gjenetik në një minimum lokal ose konvergjencën në një zgjidhje jooptimale.

4.4 Selection

Seleksionimi është faza e një algoritmi gjenetik në të cilën gjenomet individuale zgjidhen nga një popullatë për riprodhim të mëvonshëm. Në rastin tonë ne kemi përdorur Priority Queue për të bërë renditjen e kromozomeve dhe më pas me disa konstante të përzgjedhura kemi vendosur se sa prej popullsisë pasardhëse do të zgjedhet nga pjesa “më e mirë” e popullatës paraardhëse, dhe sa prej individëve “më të këqinj” të zgjedhur në mënyrë randome do të bëhen pjesë të gjeneratës së re.