

**MISKOLCI EGYETEM**

Gépészmérnöki és Informatikai Kar

Alkalmazott Matematikai Intézeti Tanszék

# SZAKDOLGOZAT



**MISKOLCI EGYETEM**

## Közlekedés modellezése ...

**Készítette:**

Kása Dániel Zoltán

Programtervező informatikus BSc

**Témavezető:**

Piller Imre, egyetemi tanársegéd

**MISKOLC, 2019**

**MISKOLCI EGYETEM**

Gépészmérnöki és Informatikai Kar

Alkalmazott Matematikai Intézeti Tanszék

**Szám:**

## **SZAKDOLGOZAT FELADAT**

Kása Dániel Zoltán (NEPTUN) programtervező informatikus.

**A szakdolgozat tárgyköre:** Szabály alapú rendszerek, számítógépi grafika, szimuláció

**A szakdolgozat címe:** Közlekedés modellezése ...

**A feladat részletezése:**

...

**Témavezető:** Piller Imre (egyetemi tanársegéd)

**A feladat kiadásának ideje:** 2018. ???.

.....  
szakfelelős

## EREDETISÉGI NYILATKOZAT

Alulírott **Kása Dániel Zoltán**; Neptun-kód: NEPTUN a Miskolci Egyetem Gépészmérnöki és Informatikai Karának végzős programtervező informatikus szakos hallgatója ezennel büntetőjogi és fegyelmi felelősségem tudatában nyilatkozom és aláírással igazolom, hogy *CÍMET ÁT KELL ÍRNI!* című szakdolgozatom saját, önálló munkám; az abban hivatkozott szakirodalom felhasználása a forráskezelés szabályai szerint történt.

Tudomásul veszem, hogy szakdolgozat esetén plágiumnak számít:

- szó szerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát megismertem, és tudomásul veszem, hogy plágium esetén szakdolgozatom visszautasításra kerül.

Miskolc, ..... év ..... hó ..... nap

.....

Hallgató

1.

szükséges (módosítás külön lapon)

A szakdolgozat feladat módosítása

nem szükséges

.....

dátum

.....

témavezető(k)

2. A feladat kidolgozását ellenőriztem:

témavezető (dátum, aláírás):

konzulens (dátum, aláírás):

.....

.....

.....

.....

.....

.....

3. A szakdolgozat beadható:

.....

dátum

.....

témavezető(k)

4. A szakdolgozat ..... szövegoldalt

..... program protokollt (listát, felhasználói leírást)

..... elektronikus adathordozót (részletezve)

.....

..... egyéb mellékletet (részletezve)

.....

tartalmaz.

.....

dátum

.....

témavezető(k)

5.

bocsátható

A szakdolgozat bírálatra

nem bocsátható

A bíráló neve: .....

.....

dátum

.....

szakfelelős

6. A szakdolgozat osztályzata

a témavezető javaslata: .....

a bíráló javaslata: .....

a szakdolgozat végleges eredménye: .....

Miskolc, .....

.....

a Záróvizsga Bizottság Elnöke

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>1</b>
<b>2. Témaköri áttekintés</b>	<b>2</b>
2.1. Hasonló célú szoftverek . . . . .	2
2.1.1. Road Traffic Library . . . . .	2
2.1.2. SUMO . . . . .	3
2.2. Meglévő térkép-adatbázisok modellje . . . . .	4
2.3. GeoSpatial adatbázisok . . . . .	5
<b>3. Matematikai modell részletezése</b>	<b>6</b>
3.1. Modell specifikálása . . . . .	6
3.2. A modellben használt elemek . . . . .	6
3.2.1. Egyenes út . . . . .	6
3.2.2. Egyirányú út . . . . .	6
3.2.3. Kanyar . . . . .	7
3.2.4. Útkereszteződés . . . . .	7
3.2.5. Személygépjármű . . . . .	7
3.2.6. Autóbusz . . . . .	7
3.2.7. Buszmegálló . . . . .	7
3.2.8. Épületek . . . . .	8
3.3. A modell célja . . . . .	8
3.3.1. Generálási szempont . . . . .	8
3.3.2. Szimulációs szempont . . . . .	8
<b>4. Generálás</b>	<b>9</b>
<b>5. Tervezés, implementáció</b>	<b>10</b>
<b>6. Szimulációk</b>	<b>11</b>
<b>7. Összegzés</b>	<b>12</b>
<b>Irodalomjegyzék</b>	<b>13</b>

# 1. fejezet

## Bevezetés

## 2. fejezet

# Témaköri áttekintés

### 2.1. Hasonló célú szoftverek

#### 2.1.1. Road Traffic Library

Az eddigi közlekedés-szimuláló szoftverek közül kiemelném először az AnyLogic által készített Road Traffic Library-t. Ezen szoftver segítségével létre lehet hozni egy úthálózatot különböző elemekből, mint például egyenes, kanyar, útkereszteződés, híd, zebra, lámpás útkereszteződés, valamint buszmegállók és parkolók. A szoftver képes különböző közlekedési szabályok alkalmazására, valamint a járművek intelligens módon választják meg a haladáshoz szükséges útvonalat. Valós-idejű szimuláció futtatására is van lehetőség, 2D-ben és 3D-ben is.



2.1. ábra. Kép a Road Traffic Library-ből

A szimulációkat felhasználva hőtérkép is előállítható, amely megmutatja mennyire voltak forgalmasak az adott területek, így egyértelművé válik hol alakul ki könnyedén

forgalmi dugó.

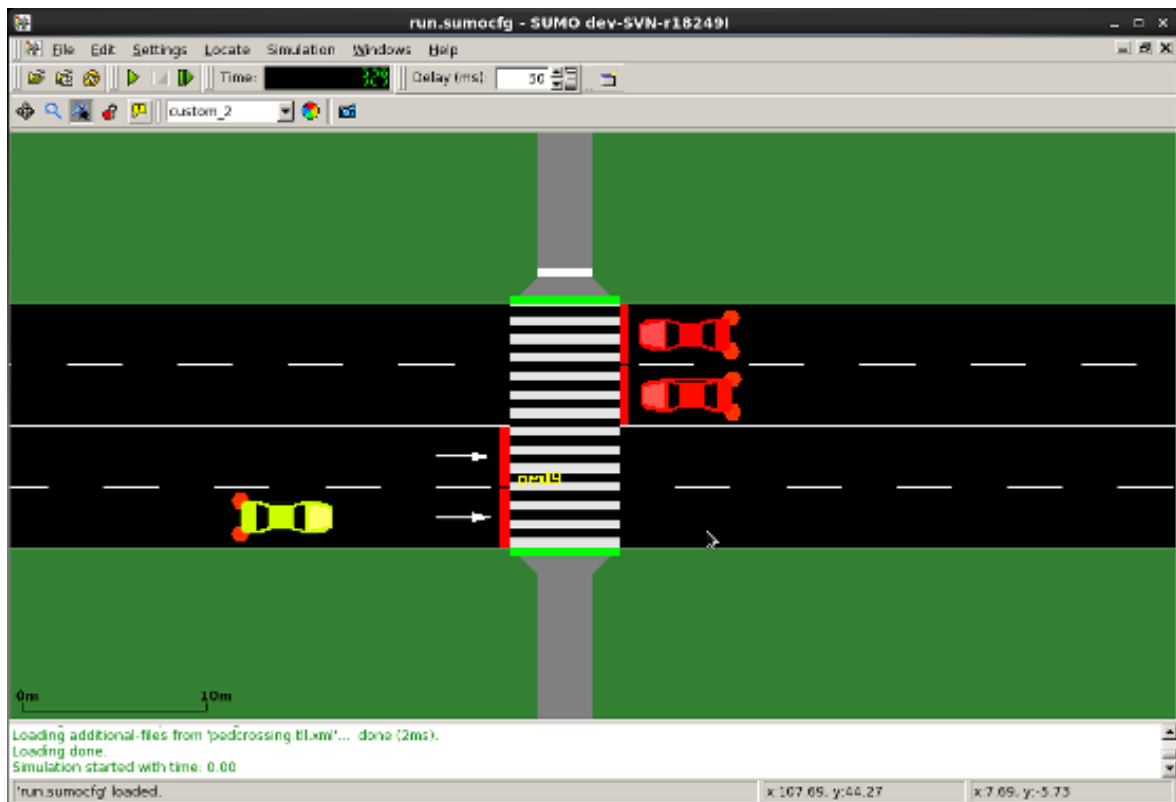
Lehetőség van a geoinformációs rendszerekben használt vektorgrafikus formátum, azaz shapefile importálására is. Ezen fájl alapján a szoftver automatikusan előállítja az úthálózatot. A szoftver bővíthető még gyalogos- és vasútforgalomra vonatkozó könyvtárakkal is.

A gyalogos könyvtár segítségével szimulálható a gyalogosforgalom sűrűsége különböző környezetekben. Ilyen környezet lehet például egy buszmegálló, vagy vasútállomás területe. Az egyes épületek felépítése is befolyásolja a forgalom sűrűségét, a gyalogosok minden akadályt, ebbe beleértve egymást is, próbálnak elkerülni.

A vasút könyvtár segítségével egy pályaudvar vasútforgalma komplex módon szimulálható. A forgalmat befolyásolja a tehervonatok rakodási ideje, a karbantartás, üzleti folyamatok lefolyása, és sok más további esemény is.

## 2.1.2. SUMO

A SUMO, azaz Simulation of Urban MObility egy ingyenes, nyílt, közlekedés-szimuláló C++-ban írt szoftver. Az első verzió 2001-ben került kiadásra, azóta már az 1.1.0-ás verziószámmal jár. Ez a szoftver lehetőséget ad több közlekedési forma (tömegközlekedés, gyalogosok, stb) modellezésére a személygépjárművek mellett. Továbbá nagy eszköztárral bír az olyanokhoz mint útvonaltervezés, káros-anyag kibocsátás, valamint vizuális megjelenítés. Lehetséges továbbá meglévő úthálózat importálása fájlból, valamint személyre szabott modellek használata. A SUMO nem tartalmaz semmilyen megkötést az úthálózat méretét, vagy az egyszerre szimulálható járművek számát illetően.



2.2. ábra. Kép a SUMO programból



---

A program lehetőséget ad a szimuláció online történő kezelésére is, a TraCI, azaz Traffic Control Interface használatával. Ilyen esetben a SUMO tulajdonképpen egy szerver, amelyet egy vagy több kliensen keresztül tudunk vezérelni. TCP alapú kliens/-szerver architektúrát használ.

A közlekedési lámpák viselkedését egyénileg be lehet állítani, vagy akár előre megírt TLS programból is lehetséges importálni. Az importáláshoz számos python szkript áll rendelkezésre. Ha nem áll rendelkezésre TLS program, a SUMO képes automatikusan generálni egyet. Ilyenkor alapértelmezett értékeket rendel minden paraméterhez amelynek értéke nem volt kapcsolókkal definiálva. Ezek közé tartoznak:

- `-tls.cycle.time` - lámpák váltási ideje.
- `-tls.yellow.time` - mennyi időre sárga a lámpa, ez alapértelmezetten az utak sebességhatára alapján történik kiszámításra.
- `-tls.minor-left.max-speed` - az útkereszteződésben balra kanyarodást megengedő sebességhatár (ha az út sebességhatára ezen érték felé esik, nem engedélyezett a balra kanyarodás.)
- `-tls.left-green.time` - ha egyszerre kap zöldet az egyenesen haladó, és a vele szemről balra kanyarodó, ezen érték idejéig a balra kanyarodó kap elsőbbséget.
- `-tls.allred.time` - minden zöld lámpa előtt ezen értéknek megfelelő ideig piros az összes lámpa (alapértelmezetten 0.)
- `-tls.default-type` - actuated értékre állítva változó hosszúságú ideig tartanak a zöld lámpák.

A szoftver alkalmas a közlekedési lámpák teljesítményének vizsgálatára, olyan útvonaltervezésre amely igyekszik a káros-anyag kibocsátást és a légszennyezést minimalizálni. A gyakorlatban történt híresebb alkalmazásai tartozik például a 2006-os labdarúgó világbajnokság-, valamint a Pápa 2005-ös Kölni látogatása során a közlekedés előrejelzése.

Maga a programcsomag tartalmazza a parancssori SUMO szoftvert, a grafikus felülettel ellátott GUISIM-et, egy úthálózat importáló NETCONVERT szoftvert, úthálózat generáló NETGEN szoftvert, egy OD2TRIPS nevű alkalmazást amely O/D mátrix alapján generál útvonalat, számos útvonal generátort, valamint egy grafikus úthálózat-szerkesztő alkalmazást.

## 2.2. Meglévő térkép-adatbázisok modellje

A Google Maps az egyik legelterjedtebb úthálózat-vizualizálásra alkalmas szoftver. Ez egy JavaScript alapú webalkalmazás, mely a geoinformációs adatok megjelenítése mellett valós-idejű forgalom elemzésre is képes.

Ehhez az adatokat többféle forrásból állítják elő. A Base Map program keretében rengeteg különböző forrásból szereztek vektoradatokat a létező úthálózatokról. Később ennek a nevét Geo Data Upload-ra váltották, viszont a lényege ugyan az maradt. Továbbá szereznek adatokat műholdképekből, Android rendszerű okostelefonok szolgáltatásai által, valamint a nemrég bevezetett "Helyi Idegenvezetők" funkcióval, amin keresztül bárki tölthet fel adatokat.

---

A valós-idejű forgalomelemzéshez kezdetekben a helyi önkormányzatok által szolgáltatott adatokat használták, melyeket az egyes helyeken felszerelt szenzorok segítségével gyűjtöttek be. Viszont ezek csak a legforgalmasabb utakról adtak információt, ezért később áttértek a "crowd-source" módszerhez. Ezzel a módszerrel egy Google Maps-et futtató okostelefon GPS adatait felhasználva ad becslést egy út forgalmosságára. Lényegében azt elemzi, mennyien küldtek GPS adatot egy adott útszakaszon azonos időben.

Az utak és más objektumok kirajzolásához a böngésző leegyszerűsített vektoradatokat kap, melyek alapján felépíti az azoknak megfelelő formákat. Minden egyes objektumhoz tartoznak különböző metaadatok, mint például a sebességhatár, haladási irány, út típusa (főút, stb), valamint egy név. Az objektumok JSON formában vannak tárolva, maga a Google Maps API pedig támogatja a GeoJSON forrásból történő adatok betöltését térképekre. Ezen információk alapján épül fel a modell amit a böngésző 2D vagy 3D formában ábrázol egy térképen.

## 2.3. GeoSpatial adatbázisok

A GeoSpatial adatbázisok adják az előbbi szolgáltatások háttérét. Ezek az adatbázisok kifejezetten a térbeli objektumokat leíró adatok tárolására, azok egyszerű lekérdezésére vannak optimalizálva. A leggyakoribb adatok pontok, vonalak, vagy poligonok formájában fordulnak elő.

A hagyományos adatbázisokhoz képest sokkal nagyobb teljesítményt nyújtanak ilyen területen, valamint gyakran az ilyen típusú adatok feldolgozásához bővíteni kell azok funkcionalitását. Ennek érdekében az Open Geospatial Consortium 1997-től kezdődően definiálta az ISO 19125 szabványt ezen funkcionalitás megvalósításához, melynek második része leírja az SQL-ben történő megvalósítást is. Az OpenGIS szabvány ugyan magába foglalja a CORBA és az OLE/COM implementációkat is, ezek nem részei az ISO szabványnak.

A szabvány többek között a következő műveleteket definiálja:

- Térbeli számítások, azaz objektumok közti távolság, vonalak hossza, stb
- Térbeli predikátumok, objektumok közötti kapcsolatokat leíró logikai lekérdezések
- Geometriai konstruktorok, alakzatot leíró adatok alapján új geometriai elem létrehozása
- Egy tetszőleges objektumhoz tartozó információk lekérdezése

## 3. fejezet

# Matematikai modell részletezése

- Modellezési probléma specifikálása - Pontosan milyen elemekből kellene felépíteni az adott város/térkép közlekedését? - Mi lenne a modell célja? - Irányított élek/egyirányú utak, többsávos utak modellje.

### 3.1. Modell specifikálása

A közlekedésre irányuló modell fő tulajdonságai közé kell hogy tartozzon az elegendő információ tárolása ahhoz, hogy alapvetően tudjanak a szimulált járművek rajta közlekedni. Ezt rengeteg féle képpen meg lehet valósítani, sokféle eleme lehet egy adott térképnek, akár minden egyes előforduló úttípusra, szabályra, táblára tekinthetünk úgy mint a modell egyedi része. Ez a megközelítés viszont lehetségesen túlbonyolítaná a modellt, ezáltal a procedurális generálás nem adna annyira elfogadható úthálózatot, lehetségesen életszerűtlen helyzetek épülhetnek fel a sok elem megléte, azok elhelyeződése miatt. Sokkal inkább célszerű a modell részeként tekinteni a főbb építőelemeket, melyek elegendőek magához az úthálózat generálásához, valamint néhány alapvető szabályt megvalósító elemet, mint például a lámpás útkereszteződés, az egyirányú út, valamint a többsávos út.

### 3.2. A modellben használt elemek

#### 3.2.1. Egyenes út

Alapvető egyenes útszakasz, ennek lehet több paramétere is. Ennek az elemnek első sorban tartalmaznia kell a rá vonatkozó sebességhatárt, valamint az úton létező sávok számát. Egyszerűen jellemezhető az elem a kezdő és a végpontjával, ugyanis az út szélessége annak paramétereiből (sávok száma) már adódik.

#### 3.2.2. Egyirányú út

Hasonló az egyenes úthoz, viszont egyértelműen meg kell határozni a két végpont közül melyik a kiindulási, melyik a célpont. Ugyan azok a megkötések vonatkozhatnak rá, mint az egyenes út szakaszra, azaz sebességkorlát, valamint sávok száma.

### 3.2.3. Kanyar

Vehető tulajdonképpen külön elemnek, de megvalósításában akár sok egymást követő egyenes szakasz, melyek közt kis folyamatos közelítések vannak a kanyar ívére. Ezen esetben elég a kanyar kezdő és végpontját, valamint a használt egyenes szakaszok számát megadni annak jellemzésére. Alternatívaként a kanyar szöge is felhasználható jellemzésre. Különleges eset ha a kanyar egy egyirányú úton történik, ilyenkor az egyetlen változtatás az, hogy a kanyar közelítésére használt útszakaszoknak egyirányúnak kell lenniük.

### 3.2.4. Útkereszteződés

Három vagy több út találkozásánál használatos elem. Tulajdonságai közé tartozhat az hogy lámpás útkereszteződés, vagy sem, valamint ha nem lámpás akkor van-e megállási kötelezettség. Lehetséges valamint egyszerű esetben, azaz betorkolló útnál eljelölni a felülrendelt és alárendelt utat is. Az útkereszteződés jellemezhető annak középpontjával, valamint az azt érintő utak halmazával.

### 3.2.5. Személygépjármű

Legalapvetőbb közlekedési jármű. Az autókról nyilván kell tartani azok jelenlegi pozícióját, ebbe beleértve hogy melyik objektumon tartózkodnak éppen, és annak pontosan milyen pontján. Továbbá azt is hogy merre tartanak éppen, lokálisan a jelenlegi útszakasz végpontját, globálisan pedig a városban azon csomópont, ahova el szeretne jutni. Ezen elem jellemzői közé tartozik a sebessége, valamint többsávos úton hanyadik sávban közlekedik éppen. Tudnia kell a vele egy úton közlekedő járművekről is, azok pozíciójától függ a viselkedése. A járművek kezdetben véletlenszerű helyen kezdenek, majd ha sikeresen eljutottak a célpontjukhoz eltűnnek a modelből (leparkolt kocsinak tekintve). Paraméterezést tekintve ilyenkor új jármű jelenik meg ha jelenleg kevesebb van mint a beállított maximális érték.

### 3.2.6. Autóbusz

Kissé különlegesebb járműtípus, vonatkozik rá néhány további szabály. Többek között elsőbbsége van megállóból elindulva, útkereszteződésnél nagy ívben kell fordulnia, tehát többsávos út esetén ilyenkor beljebb kell venni. A buszmegállók között megkötött útvonalon kell haladnia, mindegyiknél pedig meg kell állnia. A szimuláció teljes ideje alatt az útvonalat követi, kezdetben egyik véletlenül kijelölt megállóból indul.

### 3.2.7. Buszmegálló

Autóbusznak elhelyezett megállási pont. Útkereszteződésen kívül bármely csomópont eljelölhető buszmegállónak. Szükséges információ a buszmegállók közötti minimum távolság, valamint az összes buszmegálló száma. Ezek az adatok paraméterezhetőek is. Köztük az útvonal kiszámítása történhet egy véletlenszerűen választott kezdőponttól a kialakult gráf mohó algoritmussal történő bejárásával.

#### 3.2.8. Épületek

Az utak között kialakult üres területek feltöltésére szolgáló objektumok. Jellemzői közé tartozik az épület négy csúcsa, valamint a magassága. Az épület pozíciójának valamint méreteinek meghatározása a változatosság érdekében véletlenszerű.

## 3.3. A modell célja

### 3.3.1. Generálási szempont

Magának a városnak, úthálózatnak a generálására vonatkozóan a következők a célok:

- Viszonylag kevés elemből álljon össze a generált város
- Ne legyen életszerűtlen az összeállított úthálózat
- Tartalmazza a felvetett közlekedési helyzetek szimulálására szükséges elemeket
- Vizuálisan emlékeztessen egy átlagos városra
- Bizonyos mértékig paraméterezhető legyen, főleg a méreteket érintve

### 3.3.2. Szimulációs szempont

A már legenerált úthálózaton a szimulációra tekintve ezek a célok:

- A járművek kövessék a modellben definiált alapvető szabályokat
- A szimuláció bizonyos mértékben paraméterezhető legyen, főleg a járművek számát illetően
- Reagáljanak egymásra a járművek
- Ne legyen életszerűtlen a járművek viselkedése
- Teljesítmény szempontjából elfogadható legyen egy adott méretig
- Felhasználható statisztikai eredményeket le lehessen menteni

## 4. fejezet

# Generálás

- Térkép, úthálózat és egyéb elemek generálása - Alapelemek megjelenítése HTML Canvas-en. - Elsősorban gráfokról van szó.

## 5. fejezet

# Tervezés, implementáció

- Bemutatni a használt eszközkészletet. - UML osztálydiagram, szekvencia diagramok, blokkvázlat és hasonló dolgok. - JavaScript implementációval kapcsolatos tudnivalók.
- WebGL-ről részletesen írni.

## 6. fejezet

### Szimulációk

- Különböle paraméterezéseket megnézni. - Statisztikai jellegű vizsgálatokat végezni a kapott eredményekre.



## 7. fejezet

### Összegzés

# Irodalomjegyzék

- [https://www.researchgate.net/publication/228966705\\_A\\_Review\\_of\\_Traffic\\_Simulation\\_Software](https://www.researchgate.net/publication/228966705_A_Review_of_Traffic_Simulation_Software)
- <https://www.anylogic.com/road-traffic/>
- [https://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931\\_read-41000/](https://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/)

---

## CD-melléklet tartalma

A dolgozat PDF változatát a `dolgozat.pdf` fájlban találjuk.

A dolgozat L<sup>A</sup>T<sub>E</sub>X segítségével készült. A forrásfájlok a `dolgozat` jegyzékben találhatóak.

...