# 1 Úvod

V této práci je řešena implementace celulárního automatu [1], která bude použita pro sestavení modelu [1] jednoproudé okružní křižovatky (též kruhového objezdu). Na základě modelu a simulačních experimentů [1] bude ukázáno chování systému [1] v podmínkách běžného dopravního provozu. Smyslem experimentů je demonstrovat, že změnou parametrů okružní křižovatky, lze zvýšit kapacita (propustnost) těchto křižovatek.

## 1.1 Kdo se na práci podílel

### 1.1.1 Autoři

- Josef Kuchař (xkucha28)
- Matej Sirovatka (xsirov00)

### 1.1.2 Literatura

V této práci jsme neměli pomoc odborného konzultanta, proto jsme čerpali z dostupné literatury. Implementace vychází primáně z článku "A Realistic Cellular Automata Model to Simulate Traffic Flow at Urban Roundabouts" [2].

## 1.2 V jakém prostředí a za jakých podmínek probíhalo experimentální ověřování validity modelu

Validita modelu byla ověřovaná vůči datům z článku [2], ze kterého vychází naše implementace. Referenční implementace neexistuje, proto jsme se museli spolehnout na dostupná data z toho článku.

Jako druhý nezávislý zdroj referenčních dat jsme použili data z článku v časopisu Dopravní inženýrství [3].

# 2 Rozbor tématu a použitých metod/technologií

V naší práci se zaměřujeme na implementaci celulárního automatu, který je klíčový pro modelování jednoproudé okružní křižovatky, běžně známé jako kruhový objezd. Tento model, založený na metodě celulárního automatu, je navržen tak, aby odrazoval chování systému v reálných podmínkách dopravního provozu v městském prostředí.

Cílem našich experimentů je demonstrovat, jak lze prostřednictvím změn v parametrech okružních křižovatek zlepšit jejich kapacitu a propustnost. Takovéto zlepšení je klíčové pro efektivní řízení dopravy, zejména ve městských a předměstských oblastech.

Pro validaci našeho modelu jsme se spolehli na data získaná z akademického článku A Realistic Cellular Automata Model to Simulate Traffic Flow at Urban

Roundabouts [2]. Tato data poskytují důležité informace o reálném dopravním chování na okružních křižovatkách. Kromě toho jsme jako sekundární zdroj použili data z článku v časopisu Dopravní inženýrství [3], což nám umožnilo provést komplexní a vícestrannou analýzu.

### 2.1 Popis použitých postupů pro vytvoření modelu a zdůvodnění, proč jsou pro zadaný problém vhodné

Celá implementace byla vypracovaná v jazyce C++ a jeho standardní knihovny. Pro spracování argumentů příkazové řádky byla použita knihovna `getopt_long`. Bylo využito několik návrhových vzorů, jeden z nich byl například `Jedináček` (Singleton) pro ukládání nastavení simulace a statistik o průjezdech vozidel. Objektový návrh byl zvolený z důvodu jednoduchého rozšiřování a přehlednosti kódu a take jednoduchého a intuitivního mapování objektů v reálnem světe do kódu. Pro vizualizaci dat jsme použili jazyk Python a knihovnu `matplotlib` a `seaborn`.

### 2.2 Popis původu použitých metod/technologií

Veškerý kód je vlastním dílem napsaný autory této zprávy, žádné části nejsou převzaté z důvodu chybějíci implementace k námi vybrané studii. Implementace byla vypracovaná v jazyce C++ (standardu C++ 20) a jeho standardní knihovny.

Pro překlad byl použit nástroj `GNU Make`.

## 3 Koncepce

### 3.1 Modelování rychlosti vozidel

Zda-li:

- `speed` = rychlost vozidla v hodnotach buňka/krok což odpovída metrům za sekundu
- `gap` = vzdalenost vozidla od vozidla před ním, nebo od konce cesty
- `p` = pravděpodobnost kolísání rychlosti, která je převzata z článku [2] a nakalibrovaná podle reálných dat

Algoritmus pohybu vozidla můžem podle článku [2] rozdělit do 2 stavů:

1. **Volný pohyb**: V tomto kroku se rychlost vozidla odvozuje jenom od jeho rychlosti. V tomhle kroku se rychlost vozidla kalibruje pomocí následujících pravidel:
    1. **Počáteční zrychlení**:
    - If `speed` $< 2$: $speed = speed + 1$
    2. **Zrychlování na maximálni povolenou rychlost**:

- Else if `speed` $< 14$: $speed = speed + 2$

3. **Kolísání rychlosti při plynulé jízďe**:

- Else:
  - If `p` $< 0.3$: $speed = speed + 1$
  - Else if `p` $< 0.6$: $speed = speed - 1$
  - Else: $speed = speed$

4. **Speed Limiting**:

- V tomhle kroku se rychlost limituje na 16, což odpovída rychlosti ~57 km/h
- If `speed` $> 16$: $speed = 16$

2. **Synchronizovaný pohyb**

- If `gap` $<$ `speed`: $speed = gap \times \frac{2}{3}$

*Shrnutí*:

- Pravděpodobnost (`p`) ovlivňuje rychlost v případě, že je ustálená kolem maximální povolené rychlosti na dané cestě. V tom případě se rychlost mění o 1 buňku nahoru nebo dolů s pravděpodobností 0.3. S pravděpodobností 0.4 se rychlost nemění.
- Je zavedena horní mez rychlosti, která je 16 buňek/kroků. To odpovída rychlosti ~57 km/h, která sice není legální v případě silnice s limitem 50 km/h, ale značná část řidičů tuto rychlost nedodržuje úplně přesně. Hranice byla nastavena na tuto hodnotu z důvodu, že množství vozidel pohybujících se rychleji je zanedbatelné.
- V případě, že je vzdálenost mezi vozidly menší než rychlost vozidla, tak se rychlost vozidla nastaví na 2/3 vzdálenosti mezi vozidly. Tahle rovnice vychází z pravidla tzv. `1.5 second rule` popsaného v článku [2].

## 3.2 Modelováni vjezdu vozidel do okružní křižovatky

Zda-li:

- `length` = délka vozidla čekajíciho na vjezd do okružní křižovatky v buňkách (metrech)
- `nas` = komfortní vzdálenost pro vjezd do okružní křižovatky čekajíciho vozidla od vozidla za ním na kruhovém objezdu v buňkách (metrech), tahle hodnota je vybrána z gausovy distribuce s průměrem a směrodatnou odchylkou danou v článku [2]
- `space ahead` = vzdálenost čekajíciho vozidla od vozidla před ním na kruhovém objezdu v buňkách (metrech)
- `space behind` = vzdálenost čekajíciho vozidla od vozidla za ním na kruhovém objezdu v buňkách (metrech)

Algoritmus vjezdu vozidla do okružní křižovatky můžem podle článku [2] popsat následovně:

If `space ahead` <= `length` and `space behind` >= `nas`:

- Vozidlo vjede na okružní křižovatku

Else:

- Vozidlu se znovu vygeneruje `nas` a proces se opakuje v dalším kroku simulace

# 4 Architektura simulačního modelu/simulátoru

Architekura simulačního modelu se řídí objektově orientovaným návrhem, snažili jsme se každou entitu v reálnem světe, např. silnici, kruhový objezd nebo auto reprezentovat jako objekt v kódu, který enkapsuluje jeho chování.

## 4.1 Mapování abstraktního modelu do simulačního modelu

Po spuštení programu se pomocí třídy `Args` zpracují argumenty příkazové řádky. Tyto argumenty jsou uložené ve tříde `Settings`, která je implementována jako návrhový vzor Jedináček.

Po zpracování argumentů se vytvoří instance třídy `Simulation`, která se stará o samotnou simulaci. Tato třída zabaluje všechny ostatní části simulace, jako kruhový objezd, příjezdové a výstupní cesty. Všechny tyto části jsou implementovány jako objekty, které dědí od třídy `Road`. Tato třída představuje cestu, bez ohledu na její typ (kruhový objezd, příjezdovou a výtsupní cestu). Obsahuje 2 1D vektory, `mRoad` a `mNextRoad` a metodu `update`, která vykoná aktualizaci všech buněk cesty a výsledek uloží do následující cesty. Na konci kroku se pomocí funkce `applyUpdate` nahradí `mRoad` za `mNextRoad` a `mNextRoad` je vynulována.

Jednotlivé dědící třídy, např. `Roundabout`, přepisují metodu `update` a implementují vlastní logiku pro aktualizaci stavu buněk. V základním nastavení jsou ramena křižovatky 4, tedy třída `Simulation` obsahuje 9 instancí objektů derivovaných od třídy `Road`. 1 pro samotný kruhový objezd (`Roundabout`), 4 pro příjedové cesty cesty (`Approach`), 4 pro odchozí cesty (`Outgoing`). Objekty třídy `Ougoing` také zbírají statistiky o průjezdech vozidel a ukládají je do objektu `Statistics`, který je také implementován jako návrhový vzor `Jedináček`. Ten je po skončení simulace vypíše do logovacího souboru ve formátu CSV.

Jedna buňka je modelovaná třídou `Cell`. Buňka má svůj typ (výčtový typ `CellType`) a odkaz na společná metadata vozidla `CellMeta`. Metadata vozidla obsahují informace o rychlosti, jakým výjezdem vozidlo pojede z okružní křižovatky, jeho délce a typu. Délky typů vozidel jsou vybrány z článku. Díky této architektuře se dají podstatné informace o vozidle aktualizovat na jednom místě - nedochází tak k nekonzistenci stavu.

# 5 Podstata simulačních experimentů a jejich průběh

Nejprve bude pomocí experimentů ověřena validita modelu. Následně bude pomocí experimentů ukázáno, že změnou parametrů okružní křižovatky lze zvýšit její propustnost.

## 5.1 Postup experimentování

Experimenty probíhaly pomocí implementovaného simulačního nástroje. Všechny experimenty byly spuštěny na dobu 36000 kroků simulace, což odpovídá 10 hodinám reálného času. Tato doba byla zvolena na základě experimentů v článku [2], kde byla tato doba také zvolena.

## 5.2 Jednotlivé experimenty

### 5.2.1 Experiment 1

Prvním experimentem bylo ověření základní funkčnosti modelu. Byla vytvořena okružní křižovatka se 4 rameny. Bylo sledováno zda se počet průjezdů vozidel pohybuje v přijatelném rozmezí.

TODO Tabulka

Z tabulky je vidět, že počet průjezdů vozidel se pohybuje v rozmezí 2000 až 2700 vozidel za hodinu, které byly uvedeny v článku časopisu [3]. Z toho lze usoudit, že model není zcela nesmyslný a je možné jej dále použít.

### 5.2.2 Experiment 2

V tomto experimentu byl testován vliv počtu ramen okružní křižovatky na její propustnost. Byl měněn v rozmezí 3 až 6 ramen. Průměr okružní křižovatky byl přímo úměrný počtu ramen. Na každé rameno připadalo 5 metrů průměru.

TODO Graf

Z grafu je vidět, že s přidáváním ramen se snižuje propustnost na jedno rameno. Limitujícím faktorem je skutečnost, že okružní křižovatka má stále jenom jeden pruh. Model se tedy chová podle očekávání.

### 5.2.3 Experiment 3

V tomto experimentu byl testován vliv průmeru okružní křižovatky na její propustnost. byl měněn v rozmezí 20 až 60 metrů.

TODO Graf

Z grafu je vidět, že se zvyšujícím se průměrem okružní křižovatky se zvyšuje její propustnost. Tento vztah je popsán v prezentaci [4], která na kterou se odvolává

článek [2] ze kterého vychází naše implementace. Z toho lze usoudit, že je model validní.

### 5.3 Závěry experimentů

Při implementaci byly provedeny kalibrační experimenty, které zde nejsou explicitně uvedeny. Bylo provedeny 3 experimenty v situacích popsaných výše. Z experimentů lze odvodit chování systémů s dostatečnou věrohodností a experimentální prověřování modelu bylo úspěšné.

## 6 Shrnutí simulačních experimentů a závěr

Z výsledků experimentů vyplývá, že změnou vnějšího rádiusu okružní křižovatky lze zvýšit její propustnost při předpokladu, že již není dostatečně velká. Validita modelu byla úspěšně ověřena, viz kapitola ohledně experimentů. V rámci projektu vznikl nástroj, který vychází z modelu jednoproudové okružní křižovatky a byl implementován v jazyce C++ a jeho standardní knihovny. Nástroj je schopen simulačního experimentování s parametry křižovatky a výstupními daty jsou statistiky o průjezdech vozidel. Jednotlivé parametry lze měnit jednoduše pomocí argumentů příkazové řádky.

## Literatura

[1] P. Peringer and M. Hrubý, "Modelovaní a simulace." [online], 2019. Available: https://www.fit.vutbr.cz/study/courses/IMS/public/prednasky/IMS.pdf

[2] R. Wang and M. Liu, "A realistic cellular automata model to simulate traffic flow at urban roundabouts," in *Computational science – iccs 2005*, V. S. Sunderam, G. D. van Albada, P. M. A. Sloot, and J. J. Dongarra, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 420–427. Available: https://pdodds.w3.uvm.edu/teaching/courses/2009-08UVM-300/docs/others/2005/wang2005a.pdf

[3] L. Bartoš, "Poznatky z výzkumu kapacity vjezdu do okružní křižovatky." [online], 2019. Available: https://www.dopravniinzenyrstvi.cz/clanky/poznatky-z-vyzkumu-kapacity-vjezdu-do-okruzni-krizovatky/

[4] "INTRODUCTION to roundabouts." [online], 2006. Available: https://web.archive.org/web/20060221103353/http://www.rpi.edu/dept/cits/files/ops.ppt