

**Univerzita Palackého v Olomouci**  
**Přírodovědecká fakulta**  
**Katedra geoinformatiky**



Studijní program: **P1314 Geografie**  
Obor: **1302V011 Geoinformatika a kartografie**

## **Využití P systémů pro modelování prostorových jevů**

Zbyněk Janoška

Školitel: Doc. Mgr. Jiří DVORSKÝ, Ph.D.

**Olomouc 2017**

# OBSAH

<b>ÚVOD</b>	<b>3</b>
<b>1 CÍLE PRÁCE</b>	<b>4</b>
1.1 DC 1 – model transportu vozidel v síti . . . . .	4
1.1.1 DC 1.2 – softwarový nástroj . . . . .	5
1.1.2 DC 1.3 – simulace . . . . .	5
1.2 DC 2 – model transportu cestujících v systémech veřejné dopravy . .	5
1.2.1 DC 2.1 – teoretický model . . . . .	6
<b>2 REŠERŠE</b>	<b>7</b>
2.1 Úvod do membránových výpočtů . . . . .	7
2.1.1 Základní znaky P systémů . . . . .	7
2.1.2 Formální definice . . . . .	11
2.1.3 Definice výpočtů v P systémech a jejich výsledky . . . . .	12
2.2 Rozšíření P systémů . . . . .	13
2.2.1 Paralelismus a nedeterminismus . . . . .	14
2.2.2 Organizace membrán . . . . .	15
2.2.3 Reprezentace prostorových vazeb v P systémech . . . . .	16
2.2.4 Čas v P systémech . . . . .	18
2.2.5 Pohyb objektů mezi membránami a změna membránové struktury	19
2.3 Modelování dopravního proudu . . . . .	20
2.3.1 Modelování pohybu cestujících . . . . .	22
<b>3 METODY A POSTUP</b>	<b>23</b>
3.1 Transport vozidel v síti . . . . .	23
3.2 Transport cestujících v prostředcích hromadné dopravy . . . . .	25
<b>4 VLASTNÍ ŘEŠENÍ</b>	<b>27</b>
4.1 Transportní P systém se signalizací . . . . .	27
4.1.1 Formální definice . . . . .	27

4.1.2	Popis modelu . . . . .	28
4.1.3	Kapacity membrán a synapsí . . . . .	31
4.1.4	Nenadálé události . . . . .	34
4.2	Transportní P systém s událostmi . . . . .	35
4.2.1	Důkaz turingovské úplnosti . . . . .	36
4.2.2	Návrh pravidel . . . . .	40
4.2.3	Použití a varianty . . . . .	42
4.3	Softwarový nástroj . . . . .	45
4.4	Simulace: vliv kapacity křižovatky na rozpuštění fronty . . . . .	51
4.5	Simulace: šíření invazivního druhu říční sítí . . . . .	55
4.6	Simulace: šíření znečištění říční sítí . . . . .	62
4.7	Simulace: jednoduchá náhodná procházka . . . . .	65
4.8	Simulace: ruinování hráče . . . . .	70
4.9	Transportní P systém s vozidly . . . . .	75
4.9.1	Formální definice . . . . .	75
4.9.2	Popis modelu . . . . .	76
4.9.3	Použití a varianty . . . . .	78
4.9.4	Limity . . . . .	82
<b>5</b>	<b>VÝSLEDKY</b>	<b>83</b>
<b>6</b>	<b>DISKUZE</b>	<b>89</b>
<b>7</b>	<b>ZÁVĚR</b>	<b>94</b>
	<b>POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE</b>	<b>96</b>
	<b>PŘÍLOHY</b>	<b>106</b>
	Práce se softwarovým nástrojem . . . . .	106

# ÚVOD

P systémy byly představeny Păunem (Păun, 1998) jako výpočetní model, napodobující svou strukturou a chováním živou buňku. Păun založil teoretický model P systémů na pozorování, že procesy, které se dějí uvnitř živé buňky, mohou být chápány jako výpočet. Tento teoretický přístup se nazývá membránové výpočty (*membrane computing*), zatímco formalizované matematické modely spadajícího do tohoto myšlenkového rámce se nazývají *P systémy*.

Navzdory tomu, že výpočty v P systémech nepracují s aritmetickými operacemi (podrobně viz kapitola 3), lze u velkého počtu variant P systémů, včetně těch nejjednodušších, dokázat, že jsou turingovsky kompletní. Zjednodušeně lze říci, že tyto modely je možné použít k řešení libovolných výpočetních problémů.

P systémy se skládají ze tří základních částí – *membránové struktury*, *multimnožin objektů* a *pravidel*, která je zpracovávají. Pravidla mají formu připomínající zápis chemických reakcí:  $u \rightarrow v$  a popisují přepisování množin objektů na pravé a levé straně pravidla a přemísťování objektů v rámci systému.

Výzkum membránových systémů se od svého počátku zaměřil na výpočetní vlastnosti P systémů a jejich variant. Jednoduchý model, představený Păunem (Păun, 2000) lze téměř libovolně rozšiřovat a většina publikací se zaměřila právě na definici nových variant P systémů a popis jejich výpočetních vlastností – viz například (Ciobanu et al., 2006). Reálné aplikace P systémů byly v publikované literatuře zpočátku upozaděny a pokud se objevovaly, bylo to zejména v oblasti biochemie, kde P systémy nabízejí kromě výpočetní síly též značnou expresivnost. Publikace, které by použily tento výpočetní model pro simulaci systémů s prostorovým rozměrem se začaly objevovat až ke konci první dekády tohoto století, nicméně byly buď ryze teoretické a nezabývaly se modelováním reálných systémů (Besozzi et al., 2007; Barbuti et al., 2011) nebo s prostorem explicitně nepracovaly, ačkoli modelovaný systém měl prostorovou složku (Cardona et al., 2010).

Během mého zaměstnání v Centru dopravního výzkumu, v.v.i. jsem se zabýval možnou aplikací P systémů pro modelování dopravy. Systémy dopravy, ať už zboží či lidí, lze v mnohém připodobnit k transportu v buňkách a membránových systémech a P systémy poskytují formalismus, který umožňuje modely dopravy definovat a efektivně je simulovat.

# 1 CÍLE PRÁCE

Hlavním cílem práce, vypracované na Katedře geoinformatiky Univerzity Palackého v Olomouci, je sestavit model dopravy v síti založený na formalismu P systémů. Ačkoli je téma transportu v rámci P systémů intenzivně zkoumáno, nebyly publikovány žádné práce, které by se zabývaly dopravou nebo přepravou cestujících a nákladu. Téma transportu je u P systémů zkoumáno buď na úrovni teoretických modelů nebo jsou modelovány biochemické jevy, které se od dopravy v mnohém liší; zejména tím, že transport probíhá v hierarchicky uspořádaných systémech (buňkách) nebo v tkáních, kde nehraje roli vzdálenost a závisí na sousedství.

Vzhledem k relativní novosti tématu membránových výpočtů je součástí práce podrobnější popis této výpočetní metody. Rešerše literatury je rozšířena nad rámec popisu základního modelu a odkazů na literaturu a obsahuje popis variant P systémů, které jsou relevantní pro danou problematiku a zabývají se transportem nebo reprezentací prostoru a prostorových vazeb (viz kapitola 3).

Jelikož škála reálných dopravních systémů jimiž se lze zabývat je široká, zaměřuji se v této práci na dva modely, shrnuté v dílčích cílech:

## 1.1 DC 1 – model transportu vozidel v síti

Model transportu vozidel v síti je typickou úlohou modelování dopravy. V kapitole 4.1 je navržen model pohybu vozidel v síti obsahující křižovatky se signalizací. Je navržen nejprve jednoduchý model transportu v síti se signalizovanými křižovatkami a dále rozšiřující model, který umožňuje pracovat s kapacitami křižovatek a jednorázovými událostmi. Model vychází zejména z *generalizovaných komunikačních P systémů* (Verlan et al., 2008) a *tkáňovitých P systémů se stavy komunikačních kanálů* (Freund et al., 2004), které rozšiřuje tak, aby bylo možné realisticky reprezentovat transport vozidel v síti. Kapacity křižovatek umožňují regulovat počet přepravovaných objektů, které jsou ve výpočetním kroku „obslouženy“, zatímco události umožňují simulovat procesy, které jsou jednorázové a vedou k permanentním změnám systému (přerušeni komunikace, náhlé zvýšení počtu přepravovaných objektů apod.).

Je dokázáno, že navržený model je turingovsky úplný.

### 1.1.1 DC 1.2 – softwarový nástroj

Součástí práce je softwarový nástroj, který uživateli umožňuje navrhnout si libovolný P systém (dle navrženého modelu) a provést v něm výpočet. Software je řešen jako sada skriptů v jazyce **R**. Příloha práce obsahuje návod, jak se skripty pracovat a jak interpretovat výsledky.

### 1.1.2 DC 1.3 – simulace

Je provedeno několik simulací modelových systémů, které demonstrují využitelnost pro modelování transportu. Simulace ukazují použití navrženého modelu jak pro řešení zcela teoretických úloh, mezi něž patří například náhodná procházka a z ní odvozená hra *ruinování hráče*, tak i simulace transportu s reálných sítích.

Navržený model lze použít k modelování transportu v libovolných (nejen dopravních) sítích. Signalizaci křižovatek lze zobecnit jako parametr průjezdnosti komunikace, čímž se otevírá široké pole možných využití modelu. Události lze použít k simulování nenadálých jevů, jako je například náhlé zvýšení nebo snížení počtu objektů v systému, uzavření nebo otevření komunikačních kanálů apod. Nenadálé události je velice obtížné reprezentovat v rámci běžně používaných modelů, zejména těch založených na soustavách diferenciálních rovnic a řešených analyticky. Je ukázáno, že pomocí navrženého modelu je velice snadné s tímto typem jevů pracovat. Prezentované simulace ukazují možnosti využití modelu pro modelování transportu v dopravní a říční síti.

## 1.2 DC 2 – model transportu cestujících v systémech veřejné dopravy

Veřejná doprava představuje jednodušší dopravní systém, než doprava individuální. Jsou pevně dány body, jimiž cestující vstupují a vystupují ze systému (zastávky, stanice) a k transportu dochází pomocí velkokapacitních prostředků, bez nichž není pohyb v síti možný. Nejjednodušší typ těchto systémů představují lanové a podzemní dráhy. Vzhledem k tomu, že objekty v P systémech nefungují jako agenti a nedochází k rozhodování, je možné tento model abstrahovat pro simulaci pohybu zboží, materiálu, informací apod.

### 1.2.1 DC 2.1 – teoretický model

Formální model transportu cestujících v sítích veřejné dopravy navazuje na varianty P systémů, inspirované transportem látek v buňkách a buněčných strukturách (Páun a Páun, 2002; Verlan et al., 2008) a rozšiřuje je o další prvky, umožňující práci s pravděpodobnostmi (Suzuki et al., 2000), časem (Cavaliere a Zandron, 2006) a pohybem membrán v systému (Krishna a Páun, 2005).

Kromě základního modelu jsou diskutována rozšíření, nezbytná pro věrohodnou reprezentaci přepravy cestujících. Mezi tyto rozšíření patří nutnost reprezentovat trasy vozidel, trasy cestujících, časovou variabilitu v dopravní poptávce a jízdní řády. Je ukázáno, že ačkoli tyto parametry lze pomocí úprav v navrženém modelu reprezentovat, je silně omezen nedeterminismus systému a jsou předloženy argumenty, proč pro modelování silně deterministických systémů, jakými je přeprava cestujících, nejsou P systémy vhodným výpočetním modelem.

## 2 REŠERŠE

Tato kapitola sestává ze tří dílčích podkapitol. Jelikož membránové systémy nejsou obecně známým výpočetním modelem, nejprve jsou v podkapitole 2.1 uvedeny základní informace o tomto modelu. Kapitola je obsahově rozsáhlejší, než je při rešerši obvyklé, nicméně považuji za nutné základní pojmy a definice zde uvést s ohledem na srozumitelnost modelů, které budou představeny v pozdějších kapitolách. Přehledný úvod do problematiky poskytuje například (Păun, 2010), souhrnné informace lze nalézt například v monografii (Ciobanu et al., 2006). Vzhledem k neexistující české terminologii budu uvádět u některých termínů jejich anglické znění.

Podkapitola 2.2 navazuje na obecnou definici P systémů a popisuje rozšíření, která jsou relevantní pro modelování prostoru a dopravy. Dále jsou popsány principy práce s pravděpodobnostmi a časem v P systémech, neboť pro simulaci reálných jevů je obojí nezbytné.

Podkapitola 2.3 obsahuje rešerši poznání v oblasti modelování dopravního proudu a přepravy cestujících.

### 2.1 Úvod do membránových výpočtů

Membránové výpočty jsou oblastí počítačové vědy, která se snaží abstrahovat výpočetní principy z chování a struktury živých buněk, z jejich uspořádání a organizace do tkání a struktur vyššího řádu. P systémy poprvé představil (Păun, 1998) jako distribuovaný paralelní výpočetní model inspirovaný funkcí živé buňky. P systémy pracují s multimnožinami objektů, které jsou v systému lokalizovány. Objekty a pravidla, která s nimi manipulují, jsou zapouzdřeny v membránách. Ústřední roli hraje komunikace mezi jednotlivými membránami (a okolím systému).

Existuje množství variant tohoto základního schématu.

#### 2.1.1 Základní znaky P systémů

Nejvýraznějším znakem P systémů je jejich organizace do oddělených částí. Zapouzdření – existence membrán, ohraničujících určitý prostor a jeho obsah – je jedním ze základních principů života a jedna ze tří vůdčích teorií vzniku života dokonce předpokládá, že vytvoření membrán bylo prvním krokem při přechodu z neživého v živé. Až po něm následoval vznik metabolismu a sebereplikace (Capra a Luisi, 2014).



Dalším pojmem, který byl zmíněn, je *multimnožina objektů*. V živé buňce obsahují jednotlivá oddělení buňky substance (ionty, menší molekuly, makromolekuly) plovoucí v kapalném roztoku. Nelze zde najít uspořádání - vše je blízké všemu, záleží na koncentraci a množství kopií jednotlivých substancí. Abstrahujeme-li tyto principy matematicky, dostaneme množiny objektů a jejich počty - tedy *multimnožiny*. Reprezentace multimnožiny může být různá, dále v textu bude používán následující zápis: chceme-li zapsat, že jsou přítomné objekty  $a, b, c$  v množstvích 5, 2, 6, zápis multimnožiny bude ve formě řetězce  $aaaaabbcccccc$ , zjednodušeně  $a^5b^2c^6$ ; všechny permutace řetězce jsou ekvivalentní. Objekt nemusí být reprezentován jediným znakem, například  $veh$  může reprezentovat jak tři objekty  $(v, e, h)$  tak i jediný objekt  $veh$ . Množina možných objektů pro každý systém je součástí definice systému. Mezi objekty neexistuje uspořádanost (objekt  $a$  nepředchází objektu  $b$  apod.).

Biochemické reakce probíhající v odděleních buňky jsou reprezentovány pomocí *pravidel* s ohledem na reprezentaci multimnožiny. Mají formu  $u \rightarrow v$ , kde  $u$  a  $v$  reprezentují multimnožiny objektů. Příkladem pravidla může být  $aab \rightarrow abcc$ .

Při výpočtu v P systému se uplatňují následující principy:

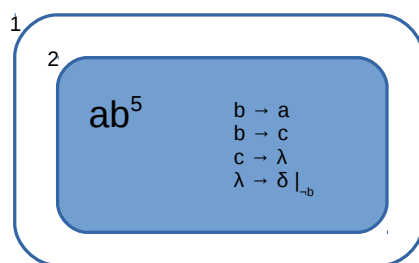
- **nedeterminismus:** obvykle je v membráně obsaženo velké množství objektů a s nimi spřaženo mnoho pravidel. Kopie objektů jsou považovány za identické a i pravidla s nimi spojená zpravidla nemají určenou prioritu. Zásadou pak je, že pravidla jsou vybírána a na objekty aplikována nedeterministicky, tedy v náhodném pořadí.
- **paralelismus:** biochemické reakce se dějí nejen nedeterministicky ale také paralelně. Pokud dvě látky mohou spolu reagovat, tak reagují nejen dvě jejich molekuly, ale všechny dostupné molekuly. V P systémech jsou v každém kroku aplikována všechna pravidla, která mohou být aplikována, na všechny dostupné objekty ve všech membránách.

P systémy jsou vnitřně diskrétní (pracují s multimnožinami objektů, jejichž počet je vyjádřen přirozenými čísly) a vyvíjejí se pomocí *pravidel*, která napodobují chemické reakce. Tato pravidla se často označují jako *evoluční*, protože při jejich aplikaci dochází k vývoji multimnožiny objektů v dané membráně. Kromě *evolučních* pravidel existují pravidla *komunikační*. *Komunikační* pravidla nepopisují změny množství objektů ( $a \rightarrow abc$ ), ale pouze přemísťují objekty v rámci membránové struktury, aniž by docházelo ke změně jejich množství nebo typu.

Výpočet probíhá v diskrétních časových krocích o stejné délce.

(Păun, 2000) definuje jako základ P systému *strukturu membrán*, která je hierarchická – membrány si lze představit jako uzly kořenového stromu, či jako Vennův diagram (obr. 1). Při znázornění formou stromu můžeme použít pojmy známé z teorie grafů – rodič/potomek, hloubka/šířka apod. Všechny membrány jsou obsaženy ve vnější membráně (*skin membrane*). Membrána, která neobsahuje žádnou další membránu, se nazývá *elementární*. Membrána ohraničuje *region*, přičemž v řadě variant P systémů lze oba pojmy membrána a region zaměňovat.

Vše, co není obsaženo v membráně, se nazývá okolím. Vnější membrána může s okolím komunikovat, ovšem není to podmínkou a řada P systémů je uzavřená.



Obrázek 1: Ukázka P systému pro výpočet hodnot šestistěnnou kostkou.

Živé buňky jsou trojrozměrnými objekty, v rámci P systémů však dimenzionalita nehraje roli. Základním principem membránových výpočtů je, že **nezáleží na pořadí, vše je blízke všemu. Důležitá je koncentrace** (Păun in (Ciobanu et al., 2006)). Pokud se více membrán nachází uvnitř jiné membrány, jejich pořadí není podstatné; to stejné platí pro objekty uzavřené v membránách. Tento princip je v přímém rozporu s prvním pravidlem geografie: „*Vše je podobné všemu, ale blízké věci jsou si podobnější*“ (Tobler, 1970). Proto P systémy, pracující s prostorem, používají jiný způsob uspořádání membrán – orientovaný graf (Barbuti et al., 2011), který může být vážený (Besozzi et al., 2007).

Uvnitř membrán se nacházejí multimnožiny objektů (*multisets*) a pravidla, která s nimi manipulují. Multimnožiny objektů reprezentují chemikálie, plovoucí uvnitř jednotlivých oddělení buňky. Opět se zde uplatňuje princip, že na pořadí a vzdálenosti nezáleží. To je důležité zejména s ohledem na nedeterminismus výpočtů, který byl zmíněn v podkapitole 2.1.1. Jednotlivé objekty se vyskytují v nenulovém a nezáporném, vždy celočíselném množství (P systémy jsou vnitřně diskrétní).

Posledním ze základních prvků membránových systémů jsou pravidla. Lze je rozřa-

dit do tří skupin podle povahy procesu, který popisují:

- Pravidla přepisující objekty v multimnožinách (*multiset-rewriting rules, evolution rules*)
- Komunikační pravidla
- Pravidla pro práci s membránami

**Pravidla přepisující objekty v multimnožinách** mají tvar  $u \rightarrow v$ , přičemž  $u$  a  $v$  jsou multimnožiny objektů.

**Příklad: aplikace pravidla** Pravidlo  $aab \rightarrow abcc$  popisuje proces, kdy spolu reagují dvě kopie objektu  $a$  s kopií objektu  $b$  za vzniku jedné kopie  $a$ , jedné kopie  $b$  a dvou kopií  $c$ . Spotřebuje se jedna kopie objektu  $a$  a vytvoří se nově dvě kopie  $c$ , přičemž počet kopií  $b$  zůstává nezměněn, ačkoli se účastní reakce. Objekt  $b$  zde funguje jako katalyzátor - reakce se účastní, ale nespotřebovává se. Použijeme-li toto pravidlo na multimnožinu  $a^5b^2c^6$ , získáme multimnožinu  $a^4b^2c^8$ .

**Komunikační pravidla** popisují proces, kdy dochází k pohybu objektů v rámci membránové struktury. Možné varianty jsou podrobněji popsány v podkapitole 2.2.5.

**Pravidla pro práci s membránami** kromě přepisování objektů manipulují i se samotnou membránovou strukturou. Detailnější popis možných rozšíření poskytuje podkapitola 2.2.5.

Na tomto místě budou uvedena dvě drobná rozšíření P systémů – rozpuštění membrány (*dissolution*) a P systémy s *promotéry/inhibitory*. Rozpuštění membrány je operace, při které dojde k zániku membrány a pravidel, která jsou s ní asociována. Všechny objekty a vnořené membrány jsou přesunuty do membrány rodičovské k rozpuštěné membráně.

Promotéry a inhibitory jsou objekty, které umožňují nebo znemožňují aplikaci pravidel, aniž by se jich samy účastnily (Bottoni et al., 2002; Ionescu a Sburlan, 2004).

**Příklad: pravidlo s inhibitem** Pravidlo s inhibitem  $u \rightarrow v|_{-b}$  je interpretováno následovně: multimnožina  $u$  se změní v multimnožinu  $v$ , pokud v membráně není přítomen objekt  $b$ .

### 2.1.2 Formální definice

Systémy založené na evolučních pravidlech (přepisujících objekty v multimnožinách) popsaných výše jsou nazývány *transition P systémy*. Formálně, *transition P systém* je konstrukt ve tvaru (Păun, 1998):

$$\Pi = (O, C, \mu, w_1, w_2, \dots, w_m, R_1, R_2, \dots, R_m, i_o)$$

kde:

1.  $O$  je konečná a neprázdná abeceda objektů,
2.  $C \subset O$  je množina katalyzátorů,
3.  $\mu$  je membránová struktura, skládající se z  $m$  membrán, označených  $1, 2, \dots, m$ ; říkáme, že membránová sktruktura (a tedy systém) je stupně  $m$ ,
4.  $w_1, w_2, \dots, w_m$  jsou řetězce z  $O$  reprezentující objekty přítomné v regionech  $1, 2, \dots, m$  membránové struktury,
5.  $R_1, R_2, \dots, R_m$  jsou pravidla asociovaná s regiony  $1, 2, \dots, m$  membránové struktury,
6.  $i_o$  je buď jeden z regionů  $1, 2, \dots, m$  a pak je tento region výstupním regionem systému (*output region*), nebo je  $0$  a pak je výsledek výpočtu uložen ve vnějším prostředí systému.

Pravidla jsou ve tvaru  $u \rightarrow v$  nebo  $u \rightarrow v\delta$ , s  $u \in O^+$  a  $v \in (O \times Tar)^*$ , kde  $Tar = \{here, in, out\}^1$ . Pravidla mohou být *kooperativní* (kde  $u$  je libovolné), *nekooperativní* (kde  $u \in O - C$ ) nebo *katalytické* (ve tvaru  $ca \rightarrow cv$  nebo  $ca \rightarrow cv\delta$ , s  $a \in O - C, c \in C, v \in ((O - C) \times Tar)^*$ ); katalyzátory se nikdy nevyvíjejí ani nemění region, jen pomáhají vývoji jiných objektů.

Membránová struktura a multimnožiny objektů v jejích odděleních určují konfiguraci P systému. Počáteční konfigurace je dána membránovou strukturou a multimnožinami objektů v jejích odděleních v době začátku výpočtu, tedy  $\mu, w_1, w_2, \dots, w_m$ . Jak multimnožiny objektů, tak membránová skruktura se v průběhu výpočtu mohou měnit. Po přidání pravidel do příslušných regionů P systému získáme kompletní reprezentaci P systému, tak jak je vidět na obrázku 1.

---

<sup>1</sup>jako  $V^*$  označujeme množinu všech řetězců z abecedy  $V$  včetně prázdné množiny  $\lambda$ , zatímco jako  $V^+$  označujeme množinu  $V^* - \lambda$  všech neprázdných řetězců z  $V$

### 2.1.3 Definice výpočtů v P systémech a jejich výsledky

Ve své základní variantě jsou P systémy synchronní zařízení ve smyslu, že jsou předpokládány globální hodiny, které určují čas pro všechny regiony systému. Během jedné časové jednotky nastane transformace (přechod, *transition*) konfigurace systému a to aplikováním pravidel v každém regionu, nedeterministicky a maximálně paralelně. Pravidla jsou vybírána náhodně a jejich použití je vyčerpávající – na konci jednoho výpočetního kroku nelze v daném regionu aplikovat žádné pravidlo. Posloupnost přechodů tvoří výpočet.

Výpočet je úspěšný, pokud zastaví (*halt*) – tedy pokud se systém dostane do stavu, kdy již není možné aplikovat žádné další pravidlo a výstupní region  $i_o$  stále existuje. Pokud je výpočet úspěšný, může být výsledek získán různými způsoby. Je-li výstupním regionem některá z vnitřních membrán, jsou počítány objekty v ní přítomné a jejich počet je výsledkem. Pokud je  $i_o = 0$  a výstupním regionem je vnější prostředí, výsledkem je počet objektů, které během výpočtu do prostředí přešly. V těchto případech je výsledkem výpočtu přirozené číslo. Pokud je rozlišováno mezi různými objekty, výsledkem je vektor čísel. Objekty, které opouštějí systém mohou být uspořádány do posloupnosti v níž opouštějí systém, v tom případě je výsledkem řetězec. Výpočty, které nezastaví, nemají žádný výsledek (nevíme, kdy je číslo spočítáno) a pokud je výstupní membrána během výpočtu rozpuštěna, výpočet je přerušen a rovněž neposkytuje výsledek. Alternativně je možné definovat množinu *terminálních* objektů  $T \subseteq O$  a počítat pouze kopie objektů z  $T$ , přičemž objekty z  $O - T$  nejsou při výstupu do vnějšího prostředí počítány.

Kvůli nedeterminismu výpočtu můžeme při jednom počátečním nastavení získat více různých výsledků. P systém počítá (generuje) množinu čísel (nebo množinu vektorů čísel, nebo jazyk).

Jsou-li P systémy použity pro simulaci a nikoli pro výpočet, zřídka dojde k zastavení systému – namísto toho je sledována konfigurace systému po dobu  $t$  časových kroků a zaznamenávána aktuální konfigurace. Takto pojaté výpočty jsou relevantní zejména při simulaci chemických (Bernardini a Manca, 2003), biochemických (Bernardini et al., 2006) nebo biologických (Pescini et al., 2006; Besozzi et al., 2007; Colomer et al., 2009) systémů. Zde navržené modely jsou určeny pro provádění simulací, nikoli výpočtů (v podkapitole 4.2.1 je nicméně podán důkaz turingovské úplnosti jednoho z navržených modelů).

Princip výpočtů je uveden na příkladu P systému simulujícího hod šestistěnnou kostkou (systém generuje číslo z množiny  $\{1;2;3;4;5;6\}$ ).

**Příklad: P systém pro výpočet hodů kostkou.** Na obrázku 1 je ukázán jednoduchý P systém, který vrací celé číslo z množiny  $\{1;2;3;4;5;6\}$  a simuluje tedy hod šestistěnnou kostkou. Systém obsahuje vnější membránu označenou jako 1, v níž je vnořena jedna elementární membrána označená jako 2. Membrána 1 neobsahuje žádné objekty ani pravidla a je definována jako výstupní region. Počet objektů, které se budou na konci výpočtu nacházet v membráně 1 je výsledkem výpočtu. Membrána 2 obsahuje multimnožinu  $ab^5$  (jeden objekt  $a$  a pět objektů  $b$ ) a následující pravidla:

1.  $b \rightarrow c$ : jedna instance objektu  $b$  se změní na jednu instanci objektu  $c$
2.  $b \rightarrow a$ : jedna instance objektu  $b$  se změní na jednu instanci objektu  $a$
3.  $c \rightarrow \lambda$ : jedna instance objektu  $c$  je spotřebována (znak  $\lambda$  označuje prázdnou množinu)
4.  $\lambda \rightarrow \delta|_{-b}$ : dojde k rozpuštění membrány, pokud v ní není přítomen objekt  $b$ . Jedná se o pravidlo s inhibitorem.

Výpočet v navrženém systému probíhá následovně:

1. V prvním výpočetním kroku není možné ve vnější membráně 1 aplikovat žádné pravidlo. V membráně 2 lze aplikovat pravidla (1.) a (2.). Pravidlo (3.) nelze aplikovat, protože se v membráně nenachází žádná instance objektu  $c$  a pravidlo (4.) nelze aplikovat, protože vyžaduje absenci objektu  $b$ , který je nyní přítomen v pěti kopiích.  
Dojde k aplikaci pravidel (1.) a (2.) na 5 instancí objektu  $b$ , pravidla jsou vybrána náhodně a jsou použita vyčerpávajícím způsobem; všech pět objektů  $b$  je spotřebováno. Na konci prvního výpočetního kroku se v membráně 2 nachází  $n \in \{1; 2; 3; 4; 5; 6\}$  kopií objektu  $a$  a  $m = 6 - n$  kopií objektu  $c$ .
2. V druhém výpočetním kroku nemohou být v membráně 1 použita pravidla (1.) a (2.) neboť všechny objekty  $b$  jsou spotřebovány. Pokud se v membráně nachází nenulový počet kopií objektu  $c$ , je na všechny kopie použito pravidlo (3.).  
Zároveň je použito pravidlo (4.), jehož aplikaci v předchozím kroku blokovala existence objektu  $b$ . Pravidlo (4.) může být použito pouze jednou, protože popisuje operaci rozpuštění membrány.  
Na konci druhého výpočetního kroku byly spotřebovány všechny objekty  $c$  použitím pravidla (3.), membrána 2 byla rozpuštěna a její obsah –  $n$  kopií objektu  $a$  byl přesunut do membrány 1.
3. Na počátku třetího výpočetního kroku existuje pouze membrána 1, v níž se nachází  $n \in \{1; 2; 3; 4; 5; 6\}$  kopií objektu  $a$ . Není možné aplikovat žádné pravidlo a výpočet úspěšně končí. Počet  $n$  kopií objektu  $a$  je výsledkem výpočtu.

## 2.2 Rozšíření P systémů

V této podkapitole budou popsána rozšíření P systémů, relevantní k modelování dopravy a reprezentaci prostoru v membránových systémech. Pro kompletní přehled dostupné literatury lze doporučit zejména webovou stránku

<http://psystems.disco.unimib.it> a také sborníky konferencí CMC (*Conference on Membrane Computing*).

### 2.2.1 Paralelismus a nedeterminismus

Základním znakem P systémů jsou paralelismus a nedeterminismus. Všechna pravidla, která mohou být aplikována jsou aplikována, ve zcela náhodném pořadí. Nedeterminismus se projevuje jak při výběru objektů tak i membrán, pokud například dochází k přemístění objektů v rámci membránové struktury. Omezení jak nedeterminismu, tak paralelismu je atraktivní zejména s ohledem pro simulaci reálných systémů, kde předpoklad absolutního nedeterminismu je často nerealistický. Existují i modely P systémů, kde aplikace pravidel probíhá asynchronně (v každém kroku může být aplikován pouze určený počet pravidel), či dokonce sekvenčně (v každém kroku je aplikováno maximálně jedno pravidlo). S těmito modely je možné se podrobně seznámit například v publikacích (Cavaliere, 2004; Dang a Ibarra, 2004; Freund, 2004, 2005).

Existuje několik způsobů, jak snížit nedeterminismus P systému. Patří mezi ně:

- **Určení priority pravidel:** pravidlo je použito jen tehdy, pokud není možné použít žádné z pravidel s vyšší prioritou.
- **Kontrola prostupnosti membrány:** membrány mohou mít měnící se tloušťku, která určuje, zda pravidlo může být vykonáno nebo ne (Madhu a Krithivasan, 2001).
- **Polarizace membrány:** v základní formě je polarizace pozitivní, negativní nebo neutrální a přes polarizovanou membránu mohou prostupovat pouze objekty s polarizací opačnou. Alternativním řešením jsou popisky membrán, které mohou nabývat obecně  $k$  stavů. Pravidla mohou být vykonána jen v membránách s daným popiskem  $k_i$ , přičemž popisky membrán mohou být v průběhu výpočtu měněny (Alhazov et al., 2004).
- **Definice cílové membrány:** namísto použití indikátorů cíle *here, out, in* (podrobně viz podkapitola 2.2.5).
- **Pravděpodobnost:** z pohledu pravděpodobnosti mají všechna pravidla stejnou šanci, že budou použita (s výjimkou těch pravidel, která být použita z různých důvodů nemohou – taková mají pravděpodobnost 0). Pravděpodobnost lze použít při práci s objekty i pravidly (Obtulowicz a Păun, 2003). Při aplikaci pravděpodobnosti na výběr pravidel lze buď vypočítat pravděpodobnost, že bude

vybráno každé z konkurenčních pravidel (např. (Suzuki et al., 2001)), nebo lze definovat pravděpodobnost pro každý pár pravidel  $r_i; r_j$ , která vyjadřuje pravděpodobnost, že po aplikování pravidla  $r_i$  bude aplikováno pravidlo  $r_j$  (Madhu, 2003). Pravděpodobnost výběru pravidla může být konstantní nebo se může v průběhu výpočtu měnit v závislosti na ostatních pravidlech a počtech objektů přítomných v membráně (Pescini et al., 2006).

Zajímavým rozšířením, které efektivně vede k omezení paralelismu, je definice spotřebované a vytvořené energie pro každé z pravidel a omezení maximálního množství energie, které může být v jednom výpočetním kroku spotřebováno (Freund, 2003).

(Besozzi et al., 2008) používají pravidla ve tvaru  $u \rightarrow (u, here)$  jako takzvaná *tichá* (*mute*) pravidla. Aplikace tichého pravidla pouze uchová instanci objektu do příštího výpočetního kroku. Efektivně je tak omezen paralelismus na úrovni objektů (některé objekty nejsou spotřebovány a jen přecházejí nezměněné do dalšího výpočetního kroku) ale je zachován paralelismus na úrovni aplikace pravidel (všechny objekty jsou zpracovány jedním z pravidel).

Většinou je však řešen opačný problém – jak dosáhnout skutečně paralelního výpočtu. Většina běžně používaných počítačů obsahuje nízký počet výpočetních jednotek a modely, které lze teoreticky řešit paralelně musí být řešeny sekvenčně. P systémy jsou modelem, který je masivně paralelní, modely mohou obsahovat desítky i stovky membrán, přičemž operace v každé z nich mohou probíhat nezávisle na operacích probíhajících v ostatních membránách. P systémy jsou proto ideálním modelem pro paralelní výpočty. Tímto tématem se vyčerpávajícím způsobem zabývá (del Amor, 2014) ve své disertační práci.

### 2.2.2 Organizace membrán

Základní organizace membrán v P systémech je hierarchická. Přírozenou modifikací je umístění membrán do sítě. Toto uspořádání imituje organizaci buněk to tkání a modely se nazývají tkáňovité P systémy (*tissue-like P systems*) (Martín-Vide et al., 2002a). V základní formě je několik (elementárních) membrán rozmístěno v prostředí a komunikují spolu za pomoci symportních/antiportních pravidel (viz podkapitola 2.2.5). Komunikace mezi membránami probíhá buď přímo – v jednom kroku, nebo nepřímo, skrz vnější prostředí. Výsledek výpočtu je zpravidla definován jako počet kopií určitého objektu v dané membráně.

Důležitou roli hraje graf v tzv. *tkáňovitých P systémech se stavy komunikačních*



*kanálů (tissue-like P systems with channel-states)* (Freund et al., 2004). Mezi dvěma membránami existuje nejvýše jedna synapse (spojení), reprezentovaná jako uspořádaná dvojice  $(i; j)$ , se kterou je asociován stav  $K$ . Aplikace pravidel, popisujících pohyb objektů mezi membránami probíhá pouze tehdy, má-li synapse  $(i; j)$  požadovaný stav  $K'$ . Stav synapse se při aplikaci pravidla může změnit. Výpočet probíhá obdobně jako v dříve zmiňovaných variantách P systémů.

Dalším rozšířením tkáňovitých P systémů jsou *populační P systémy* (Bernardini a Gheorghe, 2004). Tato varianta operuje s membránovou strukturou pomocí vazebných pravidel mezi jednotlivými membránami. Vazebná pravidla umožňují měnit synapse mezi membránami a dynamicky tak měnit membránovou strukturu v průběhu výpočtu.

### 2.2.3 Reprezentace prostorových vazeb v P systémech

Při práci s prostorem v P systémech se objevují dva přístupy. První z nich reprezentuje například práce (Besozzi et al., 2007) a je založený na tom, že jediným lokalizovaným prvkem modelu je membrána. Tento přístup lze nazvat jako **síťový**. Objekty a pravidla nejsou lokalizovány, membrány však reprezentují objekty reálného světa a jsou tudíž prostorově lokalizované – je známé umístění v rámci souřadného systému a typ geometrie, kterou reprezentují. Prostorové informace jsou pouze atributem membrán a do výpočtu nevstupují. Jsou definovány prostorové vazby mezi jednotlivými membránami – sousedství a vzdálenost, ovšem nemusí být dána absolutní poloha membránového systému nebo jeho částí. Vzdálenost je vyjádřena nepřímou, například jako pravděpodobnost nebo čas asociovaný s danou synapsí. Jak už bylo řečeno v kapitole 2.1.1, základní model P systémů s hierarchickou strukturou membrán předpokládá, že v rámci membrán neexistuje pořadí ani vzdálenost a vše je blízké všemu. Proto jsou membrány uspořádány do sítě, jako je tomu například u tkáňovitých P systémů. Každá membrána pak reprezentuje prostorově lokalizovaný objekt.

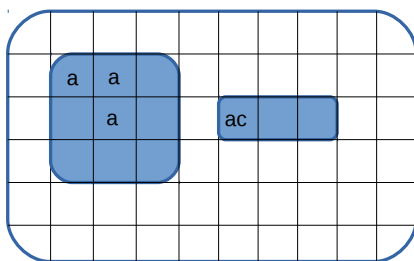
Takovýto model již dokáže popsat sousedství membrán  $i, j$ , které je vyjádřeno existencí synapse  $(i; j)$  ale nepracuje se vzdáleností. Tu je možné zahrnout při použití váh, přiřazených každé synapsi. Namísto synapsí  $(i; j)$  použijeme  $(i; j; w_{ij})$ , kde  $w \in \mathbb{R}$  vyjadřuje cenu použití synapse. Cena použití synapse může být definována jako vzdálenost mezi entitami reálného světa, které membrány reprezentují. Tento princip práce s prostorem používají například (Besozzi et al., 2007) při modelování lokálních populací. V jejich práci jsou jednotlivé geograficky oddělené populace reprezentovány membránami, mezi nimiž jsou definovány vážené synapse. Pohyb jedinců z jedné populace do druhé je omezen existencí synapse mezi membránami a pravdě-

podobnost, že jedinec z populace  $j$  bude migrovat do populace  $i$  je dán váhou  $w_{ij}$  synapse  $(i; j; w_{ij})$ . Synapse mohou být obousměrné či jednosměrné a váhy  $w_{ij}$  a  $w_{ji}$  mohou být odlišné. Každá z membrán v síti může obsahovat vnitřní membrány.

Koncepce prostoru, shrnutá v předcházejících odstavcích, umožňuje umístit membrány do prostoru, definovat mezi nimi sousedství a vzdálenost, ale předpokládá, že každá z membrán je samostatným systémem, jehož dílčí části již prostorovou složku nemají.

Druhý přístup reprezentují **prostorové P systémy** (*spatial P systems*) (Barbuti et al., 2011). Membrány jsou umístěny ve dvojrozměrném prostoru, jehož souřadnice náleží do oboru hodnot  $\mathbb{N}^2$ . Membrány mají obdélníkový tvar a mohou být do sebe vnořeny. Prostorový popis membrány je dán pomocí 2 parametrů: (1) pozicí  $p \in \mathbb{N}^2$  levého dolního rohu vzhledem k rodičovské membráně. (2) výškou  $h$  a šířkou  $w$ . Hodnoty  $w, h$  musí být nenulové. Vždy existuje vnější membrána, v níž jsou všechny membrány obsaženy. Vnější membrána má pozici  $(0, 0)$  vzhledem ke vnějšmu prostředí.

Každý objekt v membráně má rovněž svou pozici  $p$ , což znamená, že v jedné membráně se může vyskytovat více objektů stejného označení na různých pozicích (viz obrázek 2). Pravidla, která objekty zpracovávají obsahují i indikátor změny pozice.



Obrázek 2: Prostorový P systém, dle (Barbuti et al., 2011)

Model prostorových P systémů nebyl inspirován nutností práce s geografickým prostorem, jak by naznačoval název, ale nutností reprezentoval prostor v biologii, například při modelování růstu tkání (Cacciagrano et al., 2010), nebo populací ve velkém měřítku (Barbuti et al., 2014) – tedy pro reprezentaci prostoru v mnohem větších měřítcích, než s jakými se pracuje v geografii. Model dále obsahuje omezení, která z hlediska geografie nejsou opodstatněná – membrány se nemohou svými hranicemi dotýkat a objekty jsou rozděleny do dvou skupin – *běžné* a *vzájemně se vylučující*. Kromě modelování populací byly prostorové P systémy a jejich varianty použity pro simulaci růstu tkání, což je proces, který se děje v prostoru takového měřítko, s jakým

se v geografii nepracuje.

Síťový přístup reprezentace prostoru v P systémech odpovídá vektorové reprezentaci prostoru. Jednotlivé membrány si lze představit jako vektorové prvky (body, linie či polygony), zatímco prostorové P systémy odpovídají rastrové reprezentaci prostoru.

První přístup je vhodný pro prostorové modelování systémů, které mají vnitřní strukturu. Jelikož v síti umístěné membrány mohou samy obsahovat vnořené membrány, umožňují modelovat složitější systémy s vnitřní hierarchií. Populační dynamika je vhodným příkladem a byla zkoumána v rámci P systémů jak na teoretických modelech (Manca et al., 2010; Besozzi et al., 2007) tak i modelech reálných systémů podepřených terénními pozorováními (Colomer et al., 2011; Cardona et al., 2011).

Prostorové systémy se svou reprezentací prostoru podobnou rastrům byly použity rovněž pro modelování populací (Barbuti et al., 2014), ovšem mohly by najít uplatnění například při modelování šíření znečištění, kde by poskytovaly i expresivní způsob zápisu změny kvantity a kvality modelovaných látek.

Nejnovějším příspěvkem k práci s prostorem v P systémech jsou *prostorově lokalizované P systémy* (Csuhaaj-Varjú et al., 2015), které pro množinu membrán P systému definují topologické vazby, které se v průběhu výpočtu mění. Membránová struktura v této variantě nemá explicitně dané vazby mezi jednotlivými membránami (není dán graf, v jehož uzlech jsou membrány), spojení membrán je dáno až aplikací pravidel, která obsahují indikátory cílů.

#### 2.2.4 Čas v P systémech

Čas v P systémech je diskrétní. Výpočet probíhá v jednotlivých krocích, které mají stejnou délku a je předpokládána existence globálních hodin, takže čas ve všech membránách i vnějším prostředí plyne stejně. V práci (Cavaliere a Zandron, 2006) byl představen model *časovaných P systémů*, jehož hlavní myšlenkou je přiřadit každému pravidlu čas, který je potřebný pro jeho vykonání. Doba vykonání různých biochemických reakcí může mít různou délku, která nemusí být konstantní ale může záviset na stavu systému. V časovaných P systémech je každému pravidlu přiřazena konstanta  $n$ , která určuje čas vykonání pravidla. Pravidla jsou ve tvaru  $u \xrightarrow{n} v$ . Je předpokládána existence globálních hodin s uniformním intervalem, začínajícím na hodnotě 0.

Pokud je aplikováno pravidlo  $u \xrightarrow{n} v$ , v prvním kroku výpočtu dojde k odstranění multimnožiny  $u$  z membrány. Je vytvořeno *visící* (*pending*) pravidlo  $\xrightarrow{n-1} v$ . Při každém dalším kroku je hodnota  $n$  ve *visícím* pravidle snížena o 1. Jakmile hodnota  $n$

dosáhne nuly, je v membráně vytvořen objekt  $v$ .

Rozšířením tohoto modelu jsou *časované P automaty* (Barbuti et al., 2008). Časovaný P automat je časovaným automatem, kde v jednotlivých místech automatu jsou umístěny časované P systémy. Díky tomu lze efektivně modelovat navazující procesy (v případě (Barbuti et al., 2008) jsou to fáze života živočicha v průběhu roku).

Pro modelování transportu je nutné časované P systémy použít, neboť předpoklad jednotkového intervalu, ve kterém jsou vykonána všechna pravidla, je nerealistický.

## 2.2.5 Pohyb objektů mezi membránami a změna membránové struktury

Základním způsobem transportu objektů (i membrán) v P systémech je operace rozpuštění, popsána v podkapitole 2.1.1. Je to však transport jednosměrný, kdy po zániku membrán jejich obsah postupně prostupuje membránovou strukturou směrem k vnějšímu prostředí a pro modelování transportu není vhodný.

Nejjednodušším způsobem transportu, při němž nedochází ke změně membránové struktury, je použití pravidel ve tvaru  $u \rightarrow v(ind)$ , kde  $u$  a  $v$  jsou multimnožiny objektů a  $ind$  je indikátor cíle. Indikátor cíle může být prvkem z množiny (*here, in, out*), kde *here* označuje setrvání ve stávající membráně, *in* označuje přemístění do membrány, která je potomkem stávající membrány a *out* označuje přemístění objektu do membrány, která je rodičem stávající membrány. Pokud je indikátorem *in* a membrána obsahuje více vnitřních membrán, je membrána, do níž jsou objekty přemístěn, vybrána náhodně. Indikátory cíle je možné použít je u P systémů, které jsou uspořádány hierarchicky. Další možností je uvedení konkrétní membrány v indikátoru cíle bez ohledu na její pozici v rámci membránového systému (indikátor  $in_j$ , kde  $j$  je popisek některé z membrán).

(Barbuti et al., 2009) představili rozšiřující model zvaný *P systémy s transportními a difuzními membránovými kanály*. Objekty v této variantě P systémů mohou projít skrz membránu, jen pokud existuje transportní kanál, který průchod umožňuje. Existence kanálů je řízena aplikací pravidel, stejně jako evoluce objektů v membránách.

Zásadním příspěvkem ke komunikaci v membránových systémech jsou P systémy se *symportními/antiportními* pravidly (Păun a Păun, 2002). V této variantě P systémů nedochází k přepisu objektů v membránách, pouze k jejich pohybu v rámci systému. Pohyb může být realizován buď jako *symport*, kdy dva objekty prochází stěnou membrány společně stejným směrem, nebo jako *antiport*, kdy jeden objekt membránu opouští a současně do ní druhý vstupuje. Přechod osamocených objektů membráno-

vou stěnou není možný (i když *generalizované komunikační P systémy* (Verlan et al., 2008) ho umožňují a označuje se jako *uniport*). P systémy se symportem/antiportem jsou jednou z nejčastěji studovaných variant P systémů zejména díky své eleganci. Během výpočtu nedochází ke změnám objektů či jejich množství v systému, pouze změnám jejich pozice.

Alternativní model transportu představili (Martín-Vide et al., 2002b) jako *P systémy s nosiči (carriers)*. Tento systém se nejvíce blíží transportu pomocí vozidel. Nosiče jsou objekty, ke kterým mohou být přimknuty další objekty a výsledná multimnožina může měnit svou pozici v rámci membránového systému. Pravidla popisující *P systémy s nosiči* mají tvar:

- $va_1\dots a_k \rightarrow [va_1\dots a_k]$  – přimknutí objektů  $a_1\dots a_k$  k nosiči  $v$  a zapouzdření v nově vytvořené membráně
- $[va_1\dots a_k] \rightarrow va_1\dots a_k$  – odloučení objektů  $a_1\dots a_k$  a nosiče  $v$
- $[va_1\dots a_k] \rightarrow in$  – přemístění zapouzdřené membrány  $[va_1\dots a_k]$  do vnitřní membrány
- $[va_1\dots a_k] \rightarrow out$  – přemístění zapouzdřené membrány  $[va_1\dots a_k]$  do vnější membrány

Výpočet v *P systému s nosiči* probíhá tak, že k nosičům jsou přimknuty běžné objekty a výsledná multimnožina je zapouzdřena v membráně. Zapouzdřená membrána je přemístěna do vnitřní nebo vnější membrány a rozpuštěna, čímž je její obsah – objekty  $a_1\dots a_k$  a nosič  $v$  přemístěn.

Dalšími možnostmi změn membránové struktury jsou například mobilní membrány (Krishna a Păun, 2005), které jsou inspirované biologickými procesy endocytózy a exocytózy a umožňují celým membránám měnit svou pozici v rámci P systému. Zobecnujícím modelem jsou aktivní membrány (Păun, 1999; Alhazov a Ishdorj, 2004), které mohou být vytvářeny, děleny, slučovány apod. Všechny tyto varianty pracují s hierarchickou strukturou membrán, která pro modelování transportu v (dopravních) sítích není vhodná. Zajímavou aplikací aktivních membrán je simulace růstu větvičích se struktur L-systémů (Georgiou a Gheorghe, 2003; Romero-Jiménez et al., 2006).

## 2.3 Modelování dopravního proudu

Ve vyspělých zemích jsou problémy spojené s dopravou a přepravou zdrojem značných ekonomických ztrát. Efektivní rozhodování a řízení dopravy vyžaduje jasné pochopení

problematiky pohybu vozidel v síti a jejich interakcí. K těmto účelům byla během posledních více než 60 let vyvinuta řada teorií a modelů. Zájem o problematiku dopravního modelování se datuje do poloviny padesátých let minulého století, kdy byl představen model dopravy, založený na analogii pohybu vozidel v dopravním proudu a molekul kapaliny (Lighthill a Whitham, 1955). Od té doby byla navržena řada matematických modelů, popisujících dopravní proud, přičemž jednotlivé modely se lišily nejen metodologií, ale i úrovní detailu, v níž byl fenomén přepravy modelován. (Hoogendoorn a Bovy, 2001a) rozlišují tři úrovně detailu modelování.

- **(Sub-)mikroskopické modely** se zabývají modelováním individuálních entit (vozidel, řidičů) a jejich interakcí ve vysokém detailu. Modelována je například změna rychlosti v závislosti na vzdálenosti od ostatních vozidel v dopravním proudu.
- **Mezoskopické modely** se nezabývají modelováním individuálních entit (vozidel), ale popisují jejich vlastnosti, například skrz stanovení rozdělení pravděpodobností pro určité chování.
- **Makroskopické modely** popisují dopravu v agregované formě za použití nejčastěji diferenciálních rovnic. Individuální entity nejsou modelovány.

(Papageorgiou, 1998) argumentuje, že znalosti a popis dopravního proudu nikdy nedosáhne stejné úrovně, jako například Newtonovská fyzika, proto se při dopravním modelování uplatňují a nadále budou uplatňovat heuristiky a jeho cílem není nabídnout kompletní matematicko-fyzikální teorii, nýbrž model s dostatečnou schopností popisu studovaného fenoménu, přičemž pojem *dostatečný* závisí na konkrétním modelovaném jevu.

P systémy lze zařadit na pomezí modelů *mikroskopických* a *mezoskopických*. Jak bude ukázáno v kapitole 4.1, umožňují P systémy reprezentaci jednotlivých entit, ať již se jedná o vozidla či cestující (případně náklad), nicméně není možné zabývat se například změnami rychlosti vozidla, změnou jízdního pruhu apod., respektive existující modely založené na teorii celulárních automatů (Maerivoet a Moor, 2005) nebo multiagentních systémů (Bazzan a Klugl, 2009) jsou mnohem elegantnějším řešením. Naprostá většina modelů, zabývajících se modelováním dopravního proudu je navržena s ohledem na simulaci vzniku dopravních front. Používané mikroskopické i mesoskopické modely (Prigogine a Herman, 1971; Nelson, 1995; Nagel, 1996; Hoogendoorn a Bovy, 2001b) jsou zaměřeny na modelování vzniku front a pracují s hustotami vozidel, zrychlením, vzdálenostmi mezi vozidly, což jsou spojité veličiny,

které je v rámci formalismu P systémů obtížné reprezentovat. Další problém představuje diskrétnost času v P systémech. Většina modelů dopravního proudu, zejména těch založených na kinetických rovnicích, pracuje se spojitým časem.

Zásadním problémem stávajících dopravních modelů je však výpočetní náročnost (Hoogendoorn a Bovy, 2001a). Detailní modely chování vozidel či řidičů se při zvětšujícím se geografickém záběru stávají příliš komplexními. Většinou jsou proto modelovány pouze dílčí části sítě, jako jsou křižovatky (Stephanopoulos et al., 1979; Ruskin a Wang, 2002), nebo přímé úseky (Hidas, 2002), nikoli však celé sítě.

### 2.3.1 Modelování pohybu cestujících

Transport cestujících se od transportu vozidel odlišuje zejména větší volností pohybu. Zatímco u vozidel existuje jen jeden směr, kterým se lze pohybovat, a omezený počet – často jen jedna – dimenzí, které lze k pohybu použít, pohyb cestujících zahrnuje mnohem širší škálu chování. Proto bývají často používány multiagentní systémy (Enciso et al., 2016) nebo Petriho sítě (Davidrajuh a Lin, 2011). Modelování transportu cestujících se však, stejně jako modelování transportu vozidel, zaměřuje zejména na vznik front a hledání a optimalizaci *bottlenecků* (Guizzi et al., 2009; Davidrajuh a Lin, 2011; Enciso et al., 2016). Vyčerpávající přehled modelování pohybu chodců poskytuje (Daamen, 2004).

Modelování transportu cestujících v železniční či podzemní dráze bylo zkoumáno zejména s ohledem na pohyb cestujících ve stanicích (Tsang a Ho, 2004), při nástupu/výstupu (Bae et al., 2012) nebo při výběru cesty (Li a Zhu, 2016).

## 3 METODY A POSTUP

V práci jsou představeny dva modely transportu v P systémech. *Transportní P systémy se signalizací* jsou navrženy za účelem modelování pohybu vozidel v síti, která obsahuje signalizované křižovatky, ale lze jej použít k modelování transportu v sítích obecně. Model je rozšířen o kapacity křižovatek a události, které umožňují modelování jednorázových událostí.

*Transportní P systémy s nosiči* jsou navrženy pro modelování transportu cestujících hromadnou dopravou. Narozdíl od *transportních P systémů se signalizací* se přepravované objekty nepohybují mezi membránami samostatně, ale zapouzdřeny ve vozidlech, reprezenovaných jako mobilní membrány.

### 3.1 Transport vozidel v síti

Předkládaný model má za cíl modelovat chování individuálních vozidel na křižovatkách v síti, chování vozidel na přímých úsecích je ignorováno.

Membrány jsou uspořádány do orientovaného grafu, podobně jako v *generalizovaných komunikačních P systémech* (Verlan et al., 2008). Synapse mezi membránami slouží jako komunikační kanály. Přímý transport mezi membránami, které nejsou spojeny synapsí, není možný. Synapse jsou ohodnoceny váhou, která se rovná času, nutnému k přesunu mezi membránami. Synapse jsou jednosměrné a mezi dvěma membránami může existovat maximálně jedna synapse v každém směru.

Protože dopravní sítě nejsou uzavřené, musí existovat body, kterými do systému vstupují a z něj vystupují modelované objekty – vozidla. Tyto body jsou v rámci navrženého v rámci P systémů reprezentovány jako membrány. Existují tři typy membrán.

- **Černé díry:** slouží jako místa, ve kterých objekty opouštějí systém
- **Bílé díry:** slouží jako místa, kterými jsou objekty do systému přivedeny
- **Křižovatky:** jsou klasickými membránami v *tkáňovitých P systémech*, dochází zde k distribuci objektů do jiných membrán

Černé a bílé díry se nemusí nezbytně nacházet na periférii sítě, kde reprezentují okraj modelovaného prostředí. Parkoviště, residenční a komerční objekty jsou rovněž místy, chovajícími se jako bílé a černé díry a pozice těchto membrán v síti může být libovolná.



V dopravní síti mohou být křižovatky vybaveny (světelnou) signalizací, která určuje, které cesty je možné použít. Signalizaci lze v  $P$  systémech reprezentovat buď jako polarizaci membrán a umožnit transport jen objektů se shodnou polarizací (tento způsob jsme použili v dřívější práci, viz (Dvorský et al., 2012)), nebo použít princip *stavů komunikačních kanálů* (Freund et al., 2004), kde jsou synapsím spojujícím membrány přiřazeny stavy a objekty jimi mohou procházet pouze pokud jsou synapse v určeném stavu. Tento typ reprezentace je vhodnější, protože umožňuje aby z jedné membrány vedlo více otevřených komunikačních kanálů současně a je zde použit. Předkládaný model je zjednodušený v tom, že rozeznává pouze dva stavy komunikačních kanálů – otevřený a uzavřený. Oproti *tkáňovitým  $P$  systémům se stavy komunikačních kanálů* (Freund et al., 2004) nejsou používána symportní a antiportní pravidla, nýbrž pravidla uniportní (Verlan et al., 2008). Je přesouván vždy jeden objekt, jehož kvalita se během transportu může měnit (během transportu se objekt  $a$  může změnit v  $b$ ).

Vozidla jsou v navrženém modelu reprezentována jako objekty. Je možné modelovat různé druhy vozidel (osobní automobily, nákladní automobily, autobusy, jednopátá vozidla apod.). U objektů může být přítomný indikátor cíle, nesoucí informaci o následující membráně, do níž objekt směřuje.

Chování vozidel je popsáno množinou pravidel. Pravidla popisují jednak tvorbu objektů – vozidel – v bílých dírách a jejich opouštění systému v černých dírách, jednak transport objektů sítí membrán a rovněž změnu stavu komunikačních kanálů. Situace, kdy dojde například k vytvoření fronty a vozidla operativně mění vybrané cesty, navržený model neřeší.

Souhrnný popis entit reálného světa a jejich reprezentaci v rámci navrženého  $P$  systému, který označujeme jako *transportní  $P$  systém se signalizací* nabízí následující přehled:

- **Vozidla** jsou reprezentována jako objekty. Objekty obsahují indikátor cíle v obecné formě  $(in_j)$ , kde  $j$  je popisek některé z membrán.
- **Křižovatky** jsou reprezentovány jako membrány, umístěné v uzlech orientovaného grafu.
- **Signalizace křižovatek** je reprezentována stavem synapsí – hran grafu. Graf je orientovaný, dvě křižovatky mohou být spojeny dvěma synapsemi z nichž každá umožňuje transport objektů v jednom směru. Stav synapsí může být uzavřený ( $s^-$ , „červená“) nebo otevřený ( $s^+$ , „zelená“).

- **Komunikace** jsou reprezentovány synapsemi, spojujícími membrány v síti. Synapse jsou ohodnoceny časem, nutným pro průjezd a pravděpodobností výběru komunikace.
- **Pohyb vozidel** popisují pravidla. Na úrovni objektů neexistují rozhodovací procesy. Pokud je vozidlo v křižovatce, cesta kterou se vydá je vybrána na základě pravděpodobností, asociovaných s hranami grafu, reprezentujícího dopravní síť.
- **Komunikace s okolím systému** je realizována skrz membrány, sloužící jako černé a bílé díry. V černých dírách dochází k odchodu objektů do vnějšího prostředí, zatímco v bílých dírách dochází k vytváření objektů. Černé a bílé díry jsou umístěny v uzlech grafu, stejně jako membrány, reprezentující křižovatky.

### 3.2 Transport cestujících v prostředcích hromadné dopravy

Navržený model simuluje přepravu cestujících v sítích, kde probíhá přeprava prostřednictvím velkokapacitních dopravních prostředků. Typickým příkladem je doprava v systému podzemní dráhy. Systém je navržen obecně a může být použit i pro přepravu nákladu nebo další aplikace.

Membrány jsou uspořádány v síti a reprezentují stanice dopravní sítě. Jsou uspořádány do orientovaného grafu, jehož hrany jsou ohodnoceny časem, nutným k překonání vzdálenosti mezi stanicemi. Synapse jsou obousměrné a mezi dvojicí stanic může existovat maximálně jedna synapse.

Membrány reprezentující stanice slouží pro komunikaci s vnějším prostředím, jedná se tedy o bílé i černé díry zároveň. Cestující jsou reprezentováni jako objekty.

Transport v reálných dopravních systémech probíhá pomocí dopravních prostředků. Ty jsou reprezentovány jako membrány, jejichž pozice v systému se může měnit. Jedná se o mobilní membrány (Krishna a Păun, 2005). Ve stanicích mohou být mobilní membrány vytvářeny a rozpouštěny, čímž navržený systém připomíná *P systémy s aktivními membránami* (Alhazov a Ishdorj, 2004; Pan a Ishdorj, 2004) a *P systémy s nosiči* (Martín-Vide et al., 2002b).

Kapacita vozidel je vyjádřena pomocí katalyzátorů, reprezentujících prázdná místa ve vozidle. Při vytvoření obsahuje mobilní membrána tolik katalyzátorů, jaká je kapacita vozidla.

Nástup a výstup cestujících do vozidla popisují pravidla. Nástup je ve své podstatě antiportem, neboť prázdné místo ve vozidle a cestující ve stanici si vymění své pozice. Cestující, který je již ve vozidle může učinit dvě rozhodnutí: zůstat nebo vystoupit.

Tyto operace jsou popsány pomocí indikátorů (*here* znamená zůstaň, *out* znamená výstup).

Pohyb mobilních membrán v síti stanic, jejich vytváření a rozpouštění jsou popsány pravidly.

Souhrnný přehled modelovaných entit a jejich reprezentaci v rámci navrženého P systému uvádí následující výčet:

- **Cestující** jsou reprezentováni jako objekty.
- **Síť stanic/zastávek** je reprezentována pomocí membrán, umístěných do uzlů grafu. Synapse jsou ohodnoceny časem, nutným k přemístění z jedné stanice do druhé.
- **Dopravní prostředky** jsou reprezentovány pomocí mobilních membrán, které jsou přemístovány mezi membránami, reprezentujícími stanice.
- **Kapacita dopravních prostředků** je reprezentována pomocí katalyzátorů. Každá membrána reprezentující dopravní prostředek obsahuje při svém vytvoření daný počet katalyzátorů, odpovídající počtu prázdných míst. Pravidla popisující nástup a setrvání cestujících obsahují katalyzátory a mohou proto být aplikována nejvíce tolikrát, kolik katalyzátorů je v membráně přítomno.
- **Chování cestujících ve stanicích** je popsáno množinou pravidel.
- **Pohyb vozidel v síti** je popsán množinou pravidel.
- **Vstup a výstup cestujících do a ze systému** je popsán množinou pravidel.

## 4 VLASTNÍ ŘEŠENÍ

### 4.1 Transportní P systém se signalizací

#### 4.1.1 Formální definice

*Transportní P systém se signalizací* je konstrukt

$$\Pi = (O, \mu, w_1, \dots, w_m, K, syn, syn_p, (s_{(i,j)}), R, R_p, i_o), \quad (1)$$

kde

- $O$  je abeceda objektů. Objekt  $u$  může obsahovat indikátor cíle –  $u(in_j)|j \in 1, \dots, m$ , kde  $j$  je popiskem membrány,
- $\mu$  je membránová struktura, sestávající z  $m$  membrán, opatřených popisky  $1, \dots, m$ . Membrány jsou elementární a nachází se všechny přímo ve vnějším prostředí,
- $w_1, \dots, w_m$  jsou multimnožiny objektů, nacházejících se na počátku výpočtu v jednotlivých membránách  $1, \dots, m$ ,
- $K = \{s^+; s^-\}$  je množina stavů synapsí. Stav  $(s^+)$  reprezentuje synapsi, která je otevřená, zatímco  $(s^-)$  reprezentuje synapsi zavřenou,
- $syn \subseteq \{(i, j)|i, j \in \{1, \dots, m\}, i \neq j\}$ , je množina synapsí mezi membránami.
- $syn_p$  je množina visících (*pending*) synapsí,
- $(s_{(i,j)})$  jsou výchozí stavy synapsí  $(i, j) \in syn$ ,
- $R_i, \dots, R_m$  je množina pravidel asociovaných s jednotlivými membránami systému,
- $R_p$  je množina visících (*pending*) pravidel
- $i_o$  je buď jeden z regionů  $1, 2, \dots, m$  a pak je tento region výstupním regionem systému, nebo je 0 a pak je výsledek výpočtu uložen ve vnějším prostředí systému.

Tvar pravidel z množiny  $R$  je

$$\left[ u \mid \text{syn}' \right]_i \xrightarrow[s' \rightarrow s'']{t, p} \left[ v \right]_j, \quad (2)$$

kde

- $u, v \in O$  jsou objekty z abecedy  $O$ . Každé pravidlo může mít pouze jeden objekt na každé straně pravidla. Objekty mohou obsahovat indikátor cíle.
- $\left[ \ ]_{i \mid i \in 1, \dots, m}$  je membrána s popiskem  $i$ .
- $\text{syn}'$  je synapse z množiny synapsí a staven, označeným jako  $s'$ . Zápis  $[u \mid \text{syn}']$  značí, že pravidlo může být aplikováno pouze tehdy, pokud je v membráně přítomen objekt  $u$  a existuje synapse  $\text{syn}$  se stavem  $s'$ .
- $s', s'' \in K$  jsou stavy synapsí. Zápis  $s' \rightarrow s''$  značí, že během aplikace pravidla dojde ke změně stavu synapse  $\text{syn}$  z  $s'$  na  $s''$ .
- $t$  reprezentuje čas nutný k vykonání pravidla.
- $p$  reprezentuje pravděpodobnost, že bude pravidlo vybráno.

#### 4.1.2 Popis modelu

Pokud není uveden stav synapse  $s'$  a levá strana pravidla má tvar  $[u \mid \text{syn}]_i$ , pak je vyžadována existence synapse  $\text{syn}$  v libovolném stavu.

Pokud není uvedena změna stavu synapse  $s' \rightarrow s''$ , pak se stav synapse nemění.

Není-li uvedena pravděpodobnost  $p$  vykonání pravidla, mají všechna pravidla aplikovatelná v dané membráně stejnou pravděpodobnost výběru a pravidlo je vybráno nedeterministicky.

Pokud není uveden čas  $t$  vykonání pravidla, zabere jeho aplikace 1 časovou jednotku. Při aplikaci pravidel s delším časem aplikace se používá tzv. visících pravidel (Barbuti et al., 2008), jejichž aplikaci popisuje algoritmus 1.

V každém časovém kroku je nejprve provedena aplikace existujících visících pravidel a až po ní následuje aplikace běžných pravidel. Pravidlo, které obsahuje změnu stavu synapse nemůže být vykonáno, pokud je měněná synapse v množině visících synapsí (synapse je měněna jiným pravidlem). Každá synapse může být měněna pouze jedním pravidlem, současná změna synapse více pravidly není povolena.

---

**Algoritmus 1** Aplikace pravidla  $[u \mid \text{syn}' ]_i \xrightarrow{s' \rightarrow s''}^{\text{t,P}} [v ]_j$ 

---

- 1: odstraň objekt  $u$  z membrány  $i$
  - 2: vytvoř visící pravidlo  $\xrightarrow{s' \rightarrow s''}^{\text{t}-1} [v \mid \text{syn} ]_j$
  - 3: **if**  $s' \neq s''$  **then**
  - 4:   přidej synapsi  $\text{syn}$  do množiny  $\text{syn}_p$
  - 5: **end if**
  - 6: **while**  $t > 0$  **do**
  - 7:    $t = t - 1$
  - 8:   přejdi do dalšího výpočetního kroku
  - 9: **end while**
  - 10: vytvoř objekt  $v$  v membráně  $i$
  - 11: **if**  $s' \neq s''$  **then**
  - 12:   změň stav synapse  $\text{syn}$ ,  $s' \rightarrow s''$
  - 13:   odstraň synapsi  $\text{syn}$  z  $\text{syn}_p$
  - 14: **end if**
- 

**Příklad: pravidlo pro transport vozidla z membrány  $i$  do  $j$** 

Pravidlo má tvar

$$\left[ \text{veh}(in_j) \mid \text{syn}_{(i,j)}^+ \right]_i \xrightarrow{5,150} \left[ \text{veh} \right]_j, \quad (3)$$

Interpretace pravidla je následující: v membráně  $i$  se nachází objekt  $\text{veh}_{in_j}$ , jehož cílem je membrána  $j$ . Aplikace pravidla je podmíněna existencí synapse  $\text{syn}_{(i,j)}$ , která musí být otevřená (stav  $s^+$ ). Aplikace pravidla trvá 5 časových jednotek. Pseudopravděpodobnost výběru pravidla je 150. Po aplikování pravidla vznikne v membráně  $j$  objekt  $\text{veh}$ . Stav synapse  $\text{syn}_{(i,j)}$  se během výpočtu nezmění.

Významnou změnou navrženého modelu oproti ostatním variantám P systémů je, že indikátor cíle je součástí objektu, nikoli pravidla. Klasické varianty P systémů používají indikátory cíle pro popis chování objektu, např.  $u \rightarrow (v, out)$  popisuje situaci, kdy multimnožina  $u$  je změněna na multimnožinu  $v$ , která přejde do rodičovské membrány. Vzhledem k tomu, že ve zde představeném modelu mohou být synapse mezi membránami otevřené nebo uzavřené a průchod synapsí je možný, jen pokud je v otevřeném stavu, mohlo by dojít k následující situaci:

**Příklad: použití cílových indikátorů**

Na křižovatce  $i$  se nachází vozidlo  $\text{veh}$  a z křižovatky vedou dvě synapse do membrán  $j$  a  $k$ . Synapse  $(i, j)$  je otevřená, synapse  $(i, k)$  je uzavřená. V membráně  $i$  jsou přítomna pravidla  $\text{veh} \rightarrow (\text{veh}, in_j)$  a  $\text{veh} \rightarrow (\text{veh}, in_k)$  (zjednodušený zápis).

Poněvadž je synapse  $(i, k)$  uzavřená, pravidlo nelze aplikovat a v souladu s principem maximálního paralelismu musí být vykonána všechna pravidla, která vykonat lze. Proto všechny objekty  $\text{veh}$  musí být odeslány do membrány  $j$ .

Toto chování není z hlediska reprezentace dopravního proudu realistické. Ve sku-

tečnosti jsou cílové destinace vozidel vybírány předem a vozidla čekají, až budou komunikace otevřené (předpokládá se, že průjezdnost komunikací reguluje světelná či jiná signalizace).

Proto předkládaný model pracuje s indikátory cíle následujícím způsobem: Poté, co je objekt  $veh$  umístěn do membrány  $i$ , změní se na objekt  $veh(in_j)$  za předpokladu, že existuje synapse  $(i, j)$ . Na stavu synapse zde nezáleží. Pokud je synapse  $(i, j)$  uzavřená, objekt  $veh(in_j)$  setrvává v membráně  $i$ . Jakmile dojde k otevření synapse, objekt je přesunut do membrány  $j$ , kde je změněn na objekt  $veh$  bez indikátoru cíle.

Pozornost si zaslouhuje práce s pravděpodobností, jež rozhoduje o cestě vozidel sítě. Volba způsobu práce s pravděpodobností závisí na dostupných datech popisujících chování vozidel v síti. Data mohou být dostupná buď jako tzv. *OD matice* (*Origin-Destination Matrix*), která udává informaci o dopravní poptávce mezi body dopravní sítě (Peeta a Ziliaskopoulos, 2001). OD matice udává počet cestujících přepravujících se mezi dvojicí bodů v dopravní síti, ale neudává cestu, kterou cestující mezi počátkem a cílem vykonají. V případě menších uzavřených systémů mohou být tyto údaje přesně známé (např. přepravní průzkum v Pražském metru), častěji jsou však odhadovány na základě pozorovaných intenzit dopravního proudu (Sherali a Park, 2001). Intenzity dopravního proudu jsou druhým typem informace o dopravní poptávce. Namísto počtu přepravovaných entit mezi body dopravní sítě je známá jen vytíženost komunikací. Tento typ informace se získává například během celostátního sčítání dopravy.

Pokud je dostupná OD matice, je možné nalézt nejkratší cesty grafem a sestavit posloupnosti křižovatek, které musí být navštíveny během transportu cestujícího/vozidla z cílového do koncového bodu. Na základě znalosti posloupnosti projetých bodů a intenzity dopravy mezi nimi mohou být vypočteny pravděpodobnosti, že po použití synapse  $(i, j)$  bude použita synapse  $(j, k)$ , což odpovídá určení pravděpodobnosti pro každý pár pravidel  $r_i; r_j$ , že po aplikování pravidla  $r_i$  bude aplikováno pravidlo  $r_j$ . Tento přístup k práci s pravděpodobností používá (Madhu, 2003).

Pokud není OD matice známá a jsou k dispozici pouze intenzity  $I$  dopravy na jednotlivých komunikacích, lze na základě intenzity  $I_j$  vypočítat pravděpodobnost výběru komunikace  $j \in 1, \dots, n$  na křižovatce, z níž vede  $n$  komunikací jako  $p = \frac{I_j}{\sum_{i=1}^n I_i}$ . Součet pravděpodobností všech konkurujících pravidel (synapsí) je roven jedné.

Pro navržený model je zvolen druhý způsob práce s pravděpodobností, neboť pro reálné dopravní sítě je zřídka známa OD matice a pokud ano, je zpravidla odhadnuta z intenzit dopravy. Použití tohoto způsobu práce s pravděpodobností má za

následek, že není možné sledovat cestu jednoho objektu sítě. Objekty/vozidla nemají předem danou cestu, na každé křižovatce je určena následující křižovatka na základě intentit dopravy. Navržený model proto neslouží pro sledování individuálních vozidel, ale pro simulaci zatížení sítě a kapacity křižovatek, kde rozlišování mezi jednotlivými vozidly není nutné.

Sledovat cestu objektu sítě lze pouze v případech, že systém obsahuje jediný objekt, jak je ukázáno na příkladu simulace náhodné procházky v podkapitole 4.7.

### 4.1.3 Kapacity membrán a synapsí

Zásadním omezením navrženého modelu je absence kapacity. Dopravní stavby, ať již se jedná o přímé úseky komunikací nebo křižovatky, jsou stavěny s maximální kapacitou, která určuje počet vozidel, jež mohou stavbou projít za daný čas. Navržený model kontrolu naplnění kapacity neumožňuje ani u synapsí, ani u membrán.

**Kapacita v membránách** může být reprezentována pomocí katalyzátorů. V každé membráně reprezentující křižovatku je na počátku výpočtu přítomen katalyzátor  $k$  v  $n$  kopiích, kde  $n$  se rovná počtu vozidel, které mohou křižovatkou projít za jednu časovou jednotku (ve všech směrech). Obdobně mohou být katalyzátory použity pro omezení počtu vozidel, která vstupují do systému bílými děrami. Definice *transportního  $P$  systému se signalizací a kapacitami* je:

$$\Pi = (O, C, \mu, w_1, \dots, w_m, K, \text{syn}, \text{syn}_p, (s_{(i,j)}), R, R_p, i_o), \quad (4)$$

kde  $C$  je množina katalyzátorů  $C \subset O$  a význam ostatních symbolů zůstává totožný jako v *transportním  $P$  systému se signalizací*.

Pravidla mají tvar:

$$\left[ u \ k \mid \text{syn}' \right]_i \xrightarrow[s' \rightarrow s'']{t, p} \left[ v \right]_j, \quad (5)$$

kde  $u \in O$  a  $k \in C$ . Objekt  $k$  slouží jako katalyzátor. To znamená, že je nutný k tomu, aby pravidlo mohlo být aplikováno, ale nespotřebává se a nemění svou pozici v rámci systému.

Při aplikaci pravidel, která obsahují katalyzátory a jejichž doba vykonání je větší než jedna časová jednotka, je nutné vyřešit způsob práce s katalyzátorem během aplikování pravidla. Navržený model v prvním časovém kroku odstraní objekty ze zdrojové membrány a po uplynutí posledního časového kroku je umístí do cílové membrány.



Toto lze aplikovat i pro katalyzátory. Pokud tedy bude čas vykonání  $t$  časových jednotek, dojde v prvním kroku k odstranění katalyzátoru ze zdrojové membrány a po uplynutí  $t$  kroků bude katalyzátor vrácen zpět do zdrojové membrány. Druhou možností je použít katalyzátor při aplikaci pravidla a na konci prvního kroku vykonání pravidla ho vrátit do zdrojové membrány, nehledě na počet časových kroků, které vykonání pravidla zabere.

Tento druhý způsob je použit i zde, poněvadž lépe vystihuje modelovaný jev – dopravu vozidel. Pokud má křižovatka kapacitu stanovenou jako  $k$  vozidel, která mohou projet za časový interval, musí být tato kapacita dostupná v každém časovém kroku.

Aplikace pravidel probíhá dle následujícího algoritmu:

---

**Algoritmus 2** Aplikace pravidla  $[u\ k\ | \ syn']_i \xrightarrow{s' \rightarrow s''} [v]_j$

---

- 1: odstraň objekty  $u, k$  z membrány  $i$
  - 2: vytvoř visící pravidlo  $\xrightarrow{s' \rightarrow s''} [v\ | \ syn]_j$
  - 3: **if**  $s' \neq s''$  **then**
  - 4: přidej synapsi  $syn$  do množiny  $syn_p$
  - 5: **end if**
  - 6: vrať objekt  $k$  do membrány  $i$
  - 7: **while**  $t > 0$  **do**
  - 8:  $t = t - 1$
  - 9: přejdi do dalšího výpočetního kroku
  - 10: **end while**
  - 11: vytvoř objekt  $v$  v membráně  $i$
  - 12: **if**  $s' \neq s''$  **then**
  - 13: změň stav synapse  $syn, s' \rightarrow s''$
  - 14: odstraň synapsi  $syn$  z  $syn_p$
  - 15: **end if**
- 

Takto definovaný model umožňuje použít katalyzátory jednak pro řízení kapacity vozidel v křižovatkách a jednak pro řízení počtu vozidel, které vstupují do systému.

Pravidlo popisující příchod objektů z vnějšího prostředí má tvar:

$$\left[ k \right]_i \xrightarrow{t=1} \left[ u \right]_i, \quad (6)$$

kde  $k$  je katalyzátor. Počet objektů  $k$ , přítomných v jednotlivých bílých děrách udává počet vozidel, které v každém časovém kroku přijdou do systému.

Aby nebyl počet nově příchozích vozidel v systému stejný v každém kroku, lze do bílých děr umístit pravidlo

$$\left[ k \right]_i \xrightarrow{t=1} \left[ \right]_i, \quad (7)$$

keré je obdobou *tichého* pravidla  $u \rightarrow (u, \textit{here})$  (Besozzi et al., 2008). Popisuje situaci, kdy je katalyzátor použit pro aplikaci pravidla, které žádné objekty nespořehovává, nevytváří, ani nepřemistuje. Výsledný efekt je ten, že v každém kroku je v membráně vytvořeno  $l$  objektů *veh*, kde  $l$  je menší nebo rovno počtu objektů  $k$  v membráně.

Pravidlo popisující projetí vozidla křižovatkou má tvar:

$$\left[ \textit{veh}(in_j) k \mid \textit{syn}_{(i,j)}^+ \right]_i \xrightarrow{t} \left[ \textit{veh} \right]_j. \quad (8)$$

Objekt  $\textit{veh}(in_j)$ , nacházející se v membráně  $i$  se přesune do membrány  $j$  jako objekt *veh*, pokud je v membráně  $i$  volný katalyzátor a existuje synapse  $\textit{syn}_{(i,j)}$  ve stavu  $s^+$ .

Počet objektů  $k$  přítomných na počátku výpočtu v membráně udává maximální počet vozidel, která mohou membránou v jednom časovém okamžiku projet. Kapacita křižovatek se v čase nemění, není proto nutné vytvářet *tichá* pravidla tak, jako v případě bílých děr.

Pro zachycení kapacity bílých děr i křižovatek lze použít objekt  $k$ , není nutné definovat více typů katalyzátorů. Obecně je však možné definovat více objektů  $k_i \in C$ , které slouží jako katalyzátory.

**Kapacita synapsí** by mohla obdobně být reprezentována pomocí objektů  $k_{(i,j)}$ , které by byly přítomny v membráně  $i$  v  $m$  kopiích, kde  $m$  vyjadřuje maximální kapacitu synapse  $(i, j)$  a aplikace pravidla 8, popisujícího transport objektu by byla podmíněna jejich existencí.

Existovaly by tedy dva druhy katalyzátorů: objekty  $k$ , udávající kapacitu membrán a objekty  $k_{(i,j)}$ , udávající kapacitu synapsí.

Pravidlo 8 pro opuštění membrány by pak mělo tvar:

$$\left[ \textit{veh}(in_j), k_{(i,j)}, k \mid \textit{syn}_{(i,j)}^+ \right]_i \xrightarrow{t} \left[ k_{(i,j)}, k \right]_i \left[ \textit{veh} \right]_j, \quad (9)$$

Tento zápis pravidel vyžaduje úpravu algoritmu pro vykonání visících pravidel. Tak, jak je systém nyní popsán, by došlo o odstranění katalyzátorů  $k, k_{(i,j)}$  z membrány  $i$  v prvním kroku vykonání pravidla a na jeho konci by se oba objekty do membrány opět vrátily. To je správný postup s ohledem na kapacitu křižovatek, ni-

koli však s ohledem na kapacitu synapsí (komunikací), která se uvolní až poté, kdy ji vozidlo opustí. Z hlediska reprezentace dopravního proudu by muselo být s jednotlivými typy objektů pracováno následovně:

- **Běžné objekty** by byly na počátku prvního výpočetního kroku odstraněny z membrány  $i$  a na konci posledního výpočetního kroku vloženy do membrány  $j$
- **Katalyzátory křižovatek** by byly na počátku prvního výpočetního kroku odstraněny z membrány  $i$  a na konci prvního výpočetního kroku do ní vráceny.
- **Katalyzátory synapsí** by byly na počátku prvního výpočetního kroku odstraněny z membrány  $i$ , do které by byly na konci posledního výpočetního kroku vráceny.

Kapacitami synapsí se tato práce nadále nezabývá, představený model předpokládá, že kapacita synapsí je dostatečně vysoká, aby nemohla být v průběhu výpočtu překročena a je pracováno pouze s kapacitou křižovatek.

*Transportní P systému se signalizací* je zvláštním případem *transportního P systému se signalizací a kapacitami*, ve kterém pravidla neobsahují katalyzátory.

#### 4.1.4 Nenadálé události

Při modelování (nejen) dopravy je často cílem zájmu prozkoumání chování systému v případě nenadálých událostí. Může jimi být přerušování komunikace, náhlý výpadek nebo přírůstek modelovaných entit v některé části systému apod. Tyto události jsou ze své podstaty nepravidelné, často jednorázové a je obtížné je reprezentovat pomocí navrženého rámce P systémů.

Pro jejich reprezentaci je navržen *transportní P systém s událostmi*, což je *transportní P systém se signalizací a kapacitami*, rozšířený o množinu událostí  $Evt$ . Událost je dvojice  $(R, t)$ , kde  $R$  je pravidlo, definované jako běžné pravidlo v P systému a  $t \in \mathbb{N}^+$  je kladné celé číslo, označující pořadí časového kroku od počátku výpočtu, v němž je pravidlo vykonáno.

Vzhledem k tomu, že pravidla – tak jak jsou definována výše – obsahují identifikátor obou membrán, mezi kterými transport probíhá, není nutné události umísťovat do membrán, ale mohou být definovány na úrovni celého membránového systému.

## 4.2 Transportní P systém s událostmi

Je rozšířením *transportního P systému se signalizací a kapacitami*. Dále v textu bude používán termín *transportní P systém*, kterým se rozumí *transportní P systém se signalizací, kapacitami a událostmi*, který je definován jako konstrukt:

$$\Pi = (O, C, \mu, w_1, \dots, w_m, K, syn, syn_p, (s_{(i,j)})_{(i,j) \in syn}, R, R_p, Evt), \quad (10)$$

kde

- $O$  je abeceda objektů. Objekt  $u \in O$  může obsahovat indikátor cíle –  $u(in_j)|j \in 1, \dots, m$ , kde  $j$  je popisem membrány,
- $C \subset O$  je množina katalyzátorů,
- $\mu$  je membránová struktura, sestávající z  $m$  membrán, opatřených popisky  $1, \dots, m$ . Membrány jsou elementární a nachází se všechny přímo ve vnějším prostředí,
- $w_1, \dots, w_m$  jsou multimnožiny objektů, nacházejících se na počátku výpočtu v jednotlivých membránách  $1, \dots, m$ ,
- $K = \{s^+; s^-\}$  je množina stavů synapsí. Stav  $(s^+)$  reprezentuje synapsi, která je otevřená, zatímco  $(s^-)$  reprezentuje synapsi zavřenou,
- $syn \subseteq \{(i, j, t, p) | i, j \in \{1, \dots, m\}, i \neq j, t, p \in \mathbb{N}^+\}$  je množina synapsí mezi membránami. Každá synapse obsahuje informaci o čase  $t$  a pravděpodobnosti  $p$ ,
- $syn_p$  je množina visících (*pending*) synapsí,
- $s_{(i,j)}$  jsou výchozí stavy synapsí  $(i, j) \in syn$ ,
- $R = R_1, \dots, R_m$  je množina pravidel asociovaných s jednotlivými membránami systému,
- $R_p$  je množina visících (*pending*) pravidel,
- $Evt$  je množina událostí, přičemž událost je dvojice  $(R_{evt}, t)$ ,

Tvar pravidel z množiny  $R$  je

$$\left[ u, k \mid syn' \right]_i \xrightarrow[s' \rightarrow s'']{t, p} \left[ v \right]_j, \quad (11)$$

a událost z množiny  $Evt$  je dvojice  $(R_{Evt}, t)$ , kde  $R_{Evt}$  je pravidlem ve výše uvedeném zápisu a  $t$  je celé kladné číslo. Každé pravidlo může obsahovat maximálně jeden objekt  $u \in O$  a jeden katalyzátor  $k \in C$  na levé straně pravidla a maximálně jeden objekt  $v \in O$  na pravé straně pravidla.

Každá událost je vykonána v průběhu výpočtu maximálně jedenkrát a to v časovém kroku  $t$ . Pokud nejsou splněny podmínky vykonání pravidla v kroku  $t$ , pravidlo  $R_{Evt}$  není vykonáno. Právě probíhající krok označujeme jako  $t_i$ .

---

### Algoritmus 3 Aplikace události $(R_{Evt}, t)$

---

```

1:  $t_i = 0$ 
2:  $nextStep = TRUE$ 
3: while  $nextStep$  do
4:   if  $t_i = t$  then
5:     vykonej pravidlo  $R_{Evt}$ 
6:   end if
7:   proved' výpočtení krok v P systému
8:    $t_i = t_i + 1$ 
9:   aktualizuj hodnotu  $nextStep$ 
10: end while

```

---

Délka výpočtu je vždy stanovena pomocí hodnoty  $t(max)$ , označující maximální počet kroků, po které výpočet poběží. Pokud  $t(max) = \infty$ , pak výpočet běží do té doby, než zastaví klasickým způsobem (není možné aplikovat žádné další pravidlo). Pokud  $t(max) < \infty$ , pak výpočet běží po předem stanovený počet výpočetních kroků nebo do té doby, než zastaví klasickým způsobem. Pokud se systém dostane do stavu, kdy nelze aplikovat žádná další pravidla, ale množina událostí  $Evt$  obsahuje pravidla, jejichž čas vykonání je vyšší než aktuální čas systému, systém zastaví bez ohledu na nevykonané události. Stanovení času  $t(max)$  je nutné, pokud používáme model pro simulaci reálného systému a nikoli k výpočtu. V algoritmu 3 je použita proměnná  $nextStep$ , která označuje, zda byl výpočet ukončen (po dosažení  $t(max)$  nebo proto, že již nejsou žádná další pravidla, která by bylo možné v systému vykonat).

#### 4.2.1 Důkaz turingovské úplnosti

Častou metodou důkazu o turingovské úplnosti P systémů je simulace registrační pokladny. Terminologie i způsob důkazu použité v této podkapitole jsou převzaty z první kapitoly monografie (Ciobanu et al., 2006). Registrační pokladna sestává z daného počtu registrů, které mohou obsahovat jakékoli přirozené číslo a se kterými je manipulováno podle předem daných instrukcí: číslo může být zvýšeno nebo sníženo o jednotku (pokud je nenulové).

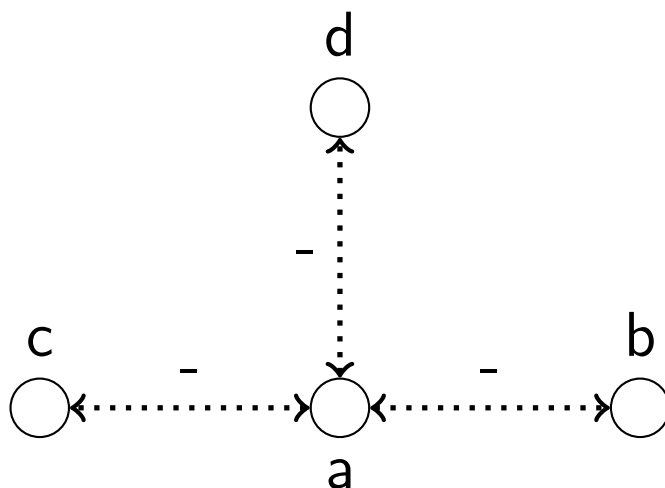
Formálně, nedeterministická registrační pokladna je konstrukt  $M = (m, B, l_0, l_h, R)$ , kde  $m \geq 1$  je počet registrů,  $B$  je konečná množina popisků instrukcí,  $l_0$  je počáteční popis,  $l_h$  je zastavující popis a  $R$  je konečná množina instrukcí popsána popisky z množiny  $B$ . Popsané instrukce mají následující formu:

- $l_1 : (add(r), l_2, l_3), 1 \leq r \leq m$  (přičti 1 k registru  $r$  a přejdi nedeterministicky na instrukci  $l_2$  nebo  $l_3$ )
- $l_1 : (sub(r), l_2, l_3), 1 \leq r \leq m$  (pokud není registr  $r$  prázdný, odečti 1 a přejdi na instrukci  $l_2$ , v opačném případě přejdi na instrukci  $l_3$ )

Registrační pokladna generuje  $k$ -rozměrný vektor přirozených čísel a je známo, že nedeterministické registrační pokladny s  $k+2$  registry jsou schopny spočítat jakoukoli množinu turingovsky spočitatelných  $k$ -rozměrných vektorů (Minsky, 1967).

P systém, který dokáže simulovat registrační pokladnu, je tudíž výpočetní silou ekvivalentní turingově stroji (Ciobanu et al., 2006).

Model registrační pokladny založený na *transportních P systémech* je zobrazen na obrázku 3. Hodnota registru je dána počtem objektů  $a_r, 1 \leq r \leq m$  přítomných v membráně  $a$ . Množina objektů  $O = \{a_r, 1 \leq r \leq m\} \cup \{l', l'' | l \in B\}$  obsahuje popisky instrukcí z množiny  $B$ , objekty  $l', l''$  a objekt  $a_r$ , jehož množství reprezentuje přirozené číslo, uložené v registru.



Obrázek 3: Membránová struktura pro simulaci registrační pokladny

Systém obsahuje 4 membrány, označené  $a, b, c, d$ . Membrána  $a$  je definována jako výstupní region. Synapse existují mezi membránami  $(a, b)$ ,  $(a, c)$ ,  $(a, d)$  a na počátku výpočtu jsou všechny ve stavu  $s^-$ . Na počátku výpočtu obsahují membrány následující objekty:

- $a$ : přítomny jsou objekty  $l_1, k$
- $b$ : přítomen je objekt  $a_r$
- $c$ : přítomny jsou objekty  $l', l''$
- $d$ : nejsou přítomny žádné objekty

Množina pravidel  $R$ , popisující systém, je :

$$\left[ l_1 \mid syn_{a,b}^- \right]_a \xrightarrow[s^- \rightarrow s^+]{t=1} \left[ l_1 \right]_b \quad (12)$$

$$\left[ l_1 \right]_b \xrightarrow{t=1} \left[ l_2 \right]_a \quad (13)$$

$$\left[ l_1 \right]_b \xrightarrow{t=1} \left[ l_3 \right]_a \quad (14)$$

$$\left[ a_r \mid syn_{a,b}^+ \right]_b \xrightarrow{t=1} \left[ a_r \right]_a \quad (15)$$

$$\left[ l_1 \mid syn_{a,c}^- \right]_a \xrightarrow[s^- \rightarrow s^+]{t=1} \left[ l_1 \right]_c \quad (16)$$

$$\left[ l' \mid syn_{a,c}^+ \right]_c \xrightarrow{t=1} \left[ l' \right]_a \quad (17)$$

$$\left[ l'' \mid syn_{a,c}^+ \right]_c \xrightarrow{t=3} \left[ l'' \right]_a \quad (18)$$

$$\left[ l' \mid syn_{a,d}^- \right]_a \xrightarrow[s^- \rightarrow s^+]{t=1} \left[ l' \right]_d \quad (19)$$

$$\left[ a_r, k \mid syn_{a,d}^+ \right]_a \xrightarrow[s^+ \rightarrow s^-]{t=1} \left[ a_r \right]_d \quad (20)$$

$$\left[ l'' \right]_a \xrightarrow{t=1} \left[ l'' \right]_d \quad (21)$$

$$\left[ l'' \mid \text{syn}_{a,d}^- \right]_d \xrightarrow{t=1} \left[ l_2 \right]_a \quad (22)$$

$$\left[ l'' \mid \text{syn}_{a,d}^+ \right]_d \xrightarrow{t=1} \left[ l_3 \right]_a \quad (23)$$

$$\left[ l_h \right]_a \xrightarrow{t=1} \left[ l_h \right]_d \quad (24)$$

Pokud je v membráně  $a$  přítomen objekt  $l_1$ , můžeme simulovat instrukce  $l_1 : (add(r), l_2, l_3)$  nebo  $l_1 : (sub(r), l_2, l_3)$ .

Simulace instrukce  $l_1 : (add(r), l_2, l_3)$  je provedena následovně:

1. Je vykonáno pravidlo 12 – objekt  $l_1$  je přesunut z membrány  $a$  do membrány  $b$ , přitom je změněn stav synapse  $(a, b)$  z  $s^-$  na  $s^+$ .
2. Ve druhém kroku je vykonáno pravidlo 15 – objekt  $a_r$  je poslán z membrány  $b$  do membrány  $a$  – hodnota registru je navýšena o 1. Aplikace pravidla je umožněna otevřením membrány  $(a, b)$ .
3. Ve druhém kroku je dále vykonáno jedno z pravidel 13, 14 – objekt  $l_1$  z membrány  $b$  je poslán do membrány  $a$  a je změněn na jeden z objektů  $l_2, l_3$ .
4. Počet objektů  $a_r$  v membráně  $a$  se navýšil o jedna a v membráně je přítomen náhodně vybraný popisek instrukce  $l_2$  nebo  $l_3$ . Tato konfigurace odpovídá korektnímu pokračování výpočtu v registrační pokladně při instrukci  $l_1 : (add(r), l_2, l_3)$ .

Simulace instrukce  $l_1 : (sub(r), l_2, l_3)$  je provedena následovně:

1. Je vykonáno pravidlo 16 – objekt  $l_1$  je přesunut z membrány  $a$  do membrány  $c$ , přitom je změněn stav synapse  $(a, c)$  z  $s^-$  na  $s^+$ .
2. Ve druhém kroku jsou vykonána pravidla 17 a 18 – objekty  $l'$  a  $l''$  jsou poslány z membrány  $c$  do membrány  $a$ . Doba vykonání pravidla 17 je jedna časová jednotka, doba vykonání pravidla 18 jsou tři časové jednotky – objekt  $l''$  se v membráně  $a$  objeví až po uplynutí tří kroků. Aplikace těchto dvou pravidel je podmíněna otevřenou synapsí  $(a, c)$ .
3. Ve třetím kroku je v membráně  $a$  přítomen objekt  $l'$  a proto je aplikováno pravidlo 19, které přesune objekt  $l'$  do membrány  $d$  a změní přitom stav synapse  $(a, d)$  z  $s^-$  na  $s^+$ .



4. Ve čtvrtém kroku, pokud je v membráně  $a$  přítomen alespoň jeden objekt  $a_r$ , je aplikováno pravidlo 21. Za přítomnosti katalyzátoru a otevřené synapse  $(a, d)$  je jedna instance objektu  $a_r$  přesunuta z membrány  $a$  do membrány  $d$ . Přitom je změněn stav synapse  $(a, d)$  z  $s^+$  na  $s^-$ .
5. V pátém kroku se v membráně  $a$  objeví objekt  $l''$  (poslaný pravidlem 18, jehož vykonání trvá 3 kroky). Pravidlem 21 je objekt  $l''$  poslán do membrány  $d$ . Na stavu synapse  $(a, d)$  nezáleží
6. V šestém kroku se v membráně  $d$  nachází objekt  $l''$ . Stav synapse  $(a, d)$  je  $s^+$ , pokud do membrány  $d$  nebyla přemístěna instance objektu  $a_r$ . V tom případě nedošlo ke snížení hodnoty registru a je použito pravidlo 22, které v membráně  $a$  vytvoří objekt  $l_2$ , reprezentující instrukci. Pokud stav synapse  $(a, d)$  je  $s^-$ , došlo k přemístění objektu  $a_r$  do membrány  $d$ , hodnota registru byla snížena o 1 a je vykonáno pravidlo 23, které v membráně  $a$  vytvoří objekt  $l_3$ , reprezentující instrukci.
7. V membráně  $a$  se nachází objekt  $l_2$ , pokud došlo ke snížení počtu objektů  $a_r$ , nebo objekt  $l_3$ , pokud se počet objektů  $a_r$  nezměnil. Tato konfigurace odpovídá korektnímu pokračování výpočtu v registrační pokladně při instrukci  $l_1 : (sub(r), l_2, l_3)$ .

Pokud je v membráně  $a$  přítomen objekt  $l_h$ , reprezentující zastavující instrukci, je použito pravidlo 24, které jej přemístí do membrány  $d$  a výpočet končí, protože není možné aplikovat žádné další pravidlo.

#### 4.2.2 Návrh pravidel

Následující pravidla popisují systém simulující pohyb vozidel v síti:

- **Příchod vozidel z vnějšího prostředí do systému.**

$$\left[ k \right]_i \xrightarrow{t=1} \left[ veh \right]_i, \quad (25)$$

Toto pravidlo se nachází pouze v membránách, které slouží jako bílé díry. Objekt  $k$  je katalyzátor a udává počet objektů  $veh$ , která v jednom časovém kroku přijdou z vnějšího prostředí do membrány. Počet objektů  $k$  je předem daný a v průběhu výpočtu se nemění.

- **Odstranění vozidel ze systému**

$$\left[ veh \right]_i \xrightarrow{t=1} \left[ \lambda \right]_i, \quad (26)$$

Toto pravidlo se nachází pouze v membránách, které slouží jako černé díry. Jeho vykonáním dojde k odstranění objektu *veh* (symbol  $\lambda$  označuje prázdnou množinu).

- **Určení cíle**

$$\left[ veh \mid syn_{(i,j)} \right]_i \xrightarrow{t=1,p} \left[ veh(in_j) \right]_i, \quad (27)$$

Toto pravidlo se nachází ve všech membránách a vozidlu bez indikátoru cíle *veh* přiřadí membránu, do níž se přesune. Podmínkou aplikace pravidla je existence synapse  $(i, j)$  v jakémkoli stavu. Jedná se o jediné pravidlo, které má určenou pravděpodobnost. Vozidlo zůstává v membráně *i*.

- **Transport mezi membránami**

$$\left[ veh(in_j), k \mid syn_{(i,j)}^+ \right]_i \xrightarrow{t} \left[ veh \right]_j, \quad (28)$$

Toto pravidlo se nachází ve všech membránách a přesune vozidlo s indikátorem cíle *j* z membrány *i*, pokud je synapse  $(i, j)$  otevřena a je přítomen katalyzátor *k*. Kapacitu membrány a tudíž maximální počet použití pravidla je dán počtem objektů *k* přítomných v membráně. Tento počet je neměnný v průběhu výpočtu.

- **Změna signalizace**

$$\left[ \lambda \mid syn_{(i,j)}^+ \right]_i \xrightarrow[t \rightarrow s^-]{t} \left[ \lambda \right]_i, \quad (29)$$

a

$$\left[ \lambda \mid syn_{(i,j)}^- \right]_i \xrightarrow[t \rightarrow s^+]{t} \left[ \lambda \right]_i, \quad (30)$$

Tato pravidla se nachází pouze v membránách, které reprezentují křižovatky a dochází v nich ke změnám signalizace. Pravidla popisují změnu stavu synapse  $(i, j)$  z otevřeného do zavřeného stavu a naopak. Ačkoli existují pouze dva stavy synapsí a bylo by teoreticky možné uvést pouze jedno pravidlo, které je bude přepínat, délka trvání jednotlivých stavů *otevřená*, *zavřená* může být různá a je proto nutné pracovat s oběma pravidly, přičemž každé může mít přiřazenou jinou časovou konstantu *t*.

Pro vykonání pravidla není nutné, aby v některé z membrán  $i, j$  byly přítomny jakékoli objekty a jak bylo uvedeno v předchozí podkapitole, existence visících synapsí zaručuje, že pravidlo nemůže být aplikováno na synapsi, jejíž stav je již měněn jiným pravidlem. Vzhledem k tomu, že mezi každou dvojicí membrán  $i, j$  mohou existovat obě synapse  $(i, j)$  a  $(j, i)$ , může být stav synapse měněn v každém ze směrů samostatně.

Tato pravidla nemusí být v systému vůbec přítomna – pak se na křižovatkách signalizace nenachází a vozidla synapsemi procházejí bez čekání (pouze otevřenými synapsemi). Pravidla pro změnu signalizace rovněž mohou být přítomna jen v některých membránách/křižovatkách a mohou regulovat otevřenost jen vybraných synapsí, vedoucích z dané membrány. Je možné mít membrány, v nichž některé synapse budou po celou dobu výpočtu otevřené, zatímco u jiných bude docházet k přepínání stavů. Ke změně stavů mohou sloužit též události.

### 4.2.3 Použití a varianty

*Transportní P systém* je navržen pro modelování chování vozidel na křižovatkách se signalizací. Chování vozidel na přímých úsecích komunikací není uvažováno. Naopak model zjednodušeně předpokládá, že překonání vzdálenosti mezi dvěma membránami zabere konstantní čas.

Namísto toho lze proměnou  $t$  asociovanou se synapsí mezi membránami chápat nikoli jako konstantu, ale jako střední hodnotu (nebo jinou charakteristiku popisující rozdělení času) a pro každý objekt, který synapsí putuje, generovat unikátní čas cesty.

Rovněž počet vozidel, která do systémů přicházejí z vnějšího prostředí, nemusí být v čase konstantní. Buď se může měnit na základě funkce, popisující změnu dopravní poptávky v čase, nebo může být předpokládána konstantní dopravní poptávka, která kolísá kolem střední hodnoty. Proměnlivý počet katalyzátorů je v rámci P systémů obtížné vyjádřit, nicméně k příchodu vozidel do systému mohou být použity události, pokud jsou definovány jako pravidlo

$$\left[ \ ]_i \xrightarrow{t=1} \left[ veh \right]_i, \quad (31)$$

Událost může být definována pro každý časový krok  $x$  krát, kde  $x$  je počet vozidel, přivedených do membrány v daném časovém kroku. Hodnota  $x$  se v čase může měnit. Pokud je příchod vozidel do systému realizován pomocí událostí, není nutné používat v bílých děrách katalyzátory.

Časová jednotka by měla být zvolena s ohledem na fakt, že chování řidičů na různých typech křižovatek se liší (Wortman a Matthias, 1983; Björklund a Åberg, 2005) a zde navržený model nedokáže rozlišit topologii křižovatek (tvar T, X, okružní křižovatka apod.) ani brát v úvahu kategorizaci komunikací (hlavní, vedlejší). Není ani možné modelovat například vyhýbací manévry vozidel při průjezdu křižovatkou, dávání přednosti nebo zastavení na značce STOP. Navíc, jak již bylo řečeno výše, jednotlivým vozidlům nejsou dány trasy a pohyb sítě se řídí pravděpodobnostmi odvozenými od intenzity dopravy, přičemž na každé křižovatce dochází k rozhodnutí, jaká bude následující stanice každého z modelovaných objektů. Nedává proto smysl sledovat pohyb jednoho objektu sítě a zvolení časového intervalu v jednotkách vteřin není vhodné. V dopravním proudu se jednotlivá vozidla následují s odstupy přibližně 2 - 15 vteřin (Kesting a Treiber, 2008). Je vhodnější zvolit větší časový interval, v jehož průběhu může křižovatkou projet i více vozidel. Není ovšem vhodné zvolit příliš velký časový interval s ohledem na změny signalizace. Pokud by byla například zvolena časová jednotka 10 minut, znamenalo by to, že se signalizace v křižovatce mění jednou za 10 minut nebo méně často. Toto v reálných dopravních sítích neplatí, signalizace se mění častěji (Urbanik et al., 2014).

Pokud je objekt transportován skrz membránu a všechny synapse jsou otevřené, stráví objekt v membráně právě jednu časovou jednotku.

Výše zmíněná doporučení platí, pokud je model použit pro simulaci pohybu vozidel v síti. Definice modelu je však obecná a je možné jej použít pro jakoukoli simulaci transportu objektů v síti. Namísto pohybu vozidel může být simulován pohyb zboží, látek nebo informací. Navržený model například umožňuje simulaci analogového počítače MONIAC (Ng et al., 2007), který byl zkonstruován jako hydraulický systém, v němž nádrže s kapalinou reprezentovaly jednotlivé aspekty národní ekonomie (vzdělávání, zdravotní péče) a odpouštěním a napouštěním vody byly simulovány peněžní operace (utrácení, výběr daní). Ačkoli se z dnešního pohledu jedná o kuriozitu, je to systém, jež v principu odpovídá zde navrženému modelu.

Transportní P systémy se signalizací mohou být použity pro simulaci průchodu sítě. V takovém případě je vhodné neurčovat pro každý objekt následující stanici na základě existence synapse, ale na základě existence otevřené synapse.

Namísto pravidel 27 a 34 by bylo použito pravidlo

$$\left[ veh, k \mid syn_{(i,j)}^+ \right]_i \xrightarrow{t} \left[ veh \right]_j, \quad (32)$$

Z výpočtu by byl zcela odstraněny cílové indikátory objektů, namísto toho by byly objekty ihned přesouvány do membrán, ke kterým vede otevřená synapse. Takto definovaný systém může stále obsahovat pravidla popisující změnu stavů synapsí. Chování přepravovaných objektů v takto pozměněném systému však mnohem více odpovídá pohybu například vody potrubím, než dopravnímu proudu.

Pokud je použita tato varianta modelu, pak dává smysl sledovat individuální objekty a posloupnost membrán, kterými projdou, než se dostanou z bílé díry do díry černé.

Pro teoretické modely pohybu v síti je možné nepracovat s pravděpodobností a zkoumat pohyb v sítích za předpokladu, že synapse jsou vybírány zcela nedeterministicky. Pak se model blíží *tkáňovitým P systémům se stavy komunikačních kanálů* (Freund et al., 2004) s tím rozdílem, že transport synapsemi trvá  $t$  časových kroků a namísto symportních/antiportních pravidel je použit uniport.

Aplikace navrženého modelu se nabízí například při zkoumání disperze populací. Membrány mohou být rozmístěny v prostoru a mít přiřazeny souřadnice. Synapse mezi nimi mohou nést informaci o času potřebném k překonání vzdálenosti. Membrány mohou být rozmístěny i do pravidelné mřížky (podobně jako v prostorových P systémech (Barbuti et al., 2011)) a model může být použit pro simulaci pohybu po povrchu. Cena pohybu po povrchu může být reprezentována jako rozdílný čas asociovaný s jednotlivými synapsemi (v obou směrech) a volba zvolené cesty pomocí pravděpodobností, které mohou být stanoveny například jako inverzní hodnota výškového gradientu. Membrány mohou být umístěny v nepravidelné či pravidelné trojúhelníkové, čtvercové či hexagonální mřížce a synapse mohou spojit  $n$  nejbližších membrán.

Představený model však nemůže být použit k modelování růstu. Každé pravidlo může obsahovat pouze jeden objekt na každé straně pravidla a nelze tak zachytit proces množení (který lze v klasických P systémech reprezentovat jako  $a \rightarrow aa$ ). Zde navržený model slouží pouze k modelování transportu jednotlivých objektů v síti za použití signalizovaných synapsí. V oblasti teorie grafu lze například model použít k modelování náhodné procházky.

Bílé a černé díry umožňují komunikaci s vnějším okolím, která je nutná pro simulaci reálné automobilové dopravy, pro řadu teoretických úloh však nemusí být černé a bílé díry použity. Všechny objekty *veh* mohou být například na počátku umístěny v jedné z membrán a může být zkoumán jejich pohyb sítí z počátečního místa. Tento

model by bylo možné použít například pro modelování šíření znečištění či infekce z bodového zdroje.

Je snadné rozšířit model tak, aby na levé i pravé straně pravidel mohl být větší počet objektů než jen jeden. Pro aplikaci v jiných oblastech než je transport vozidel/látek sítí jsou tato rozšíření nezbytná.

### 4.3 Softwarový nástroj

Pro účely simulací v navrženém modelu *transportních P systémů se signalizací* byl vytvořen softwarový nástroj v programovacím jazyku **R**. Jedná se o sadu skriptů, stažitelných ze stránek [https://github.com/janoskaz/trans\\_p\\_systems](https://github.com/janoskaz/trans_p_systems).

Nástroj umožňuje konfigurovat libovolný membránový systém, který odpovídá modelu představenému v této práci a provést v něm výpočet. Podrobný popis práce se softwarovým nástrojem je obsažen v příloze této práce, zde bude popsán pouze algoritmus výpočtu.

Základní funkcí, provádějící vlastní výpočet je funkce `simulate(membraneSystem, time=Inf)`. Funkce akceptuje dva parametry:

- *membraneSystem*, což je objekt třídy `membraneStructure`,
- *time*, udávající počet časových jednotek, po které simulace běží,

Vytvoření objektu třídy `membraneStructure` je popsáno v příloze.

Parametr *time* je celé číslo větší než 0. Pokud není parametr stanoven, je jeho hodnota nastavena na konstantu `Inf`, která v jazyce **R** reprezentuje nejvyšší možné číslo.

Funkce vrací seznam seznamů objektů v membránách systému na konci všech výpočetních kroků. Každý seznam objektů popisuje obsah membrán na konci časového kroku. Seznam objektů obsahuje katalyzátory i běžné objekty a to včetně jejich indikátoru cíle. Seznam pravidel, visících pravidel, visících synapsí a událostí není součástí výstupu.

Algoritmus výpočtu funkce `simulate` je následující:

---

**Algoritmus 4** Výpočet funkce `simulate(membraneSystem, time)`

---

```
1: ms << -membraneSystem
2: output = new list()
3: counter = 1
4: repeat
5:   rulesApplied << -FALSE
6:   applyEvents(counter) {aplikuj události}
7:   applyRules() {aplikuj pravidla v membránách}
8:   applyPendingRules() {aplikuj visící pravidla}
9:   output.push(aktuální konfigurace) {zaznamenej aktuální konfiguraci}
10:  if rulesApplied = FALSE then
11:    RETURN output
12:  end if
13:  counter = counter + 1
14: until counter = time
15: RETURN output
```

---

Znak `<< -` označuje přiřazení globální proměnné. Během běhu simulace existují dvě globální proměnné – `ms` a `rulesApplied`. Proměnná `ms` ukládá aktuální konfiguraci membránového systému a proměnná `rulesApplied` ukládá informaci o tom, zda bylo vykonáno pravidlo. `rulesApplied` je na počátku každého výpočetního kroku nastavena na `FALSE` a pokud má tuto hodnotu i na konci výpočetního kroku, je výpočet ukončen, neboť není v celém systému pravidlo, které by bylo možné vykonat.

Symbol `@` je použit k označení vlastnosti objektu. Například `rule@time` označuje vlastnost `time` objektu `rule`.

Dokud je možné v systému vykonat alespoň jedno pravidlo (ať již se jedná o pravidlo v membráně, visící pravidlo nebo událost), pokračuje výpočet do dalšího kroku. V okamžiku, kdy je aktuální krok (`counter`) rovný parametru `time`, je výpočet ukončen. Pokud se systém dostane do stavu, kdy není možné použít žádné pravidlo a hodnota `counter` je nižší než hodnota `time`, výpočet je ukončen. Přítomnost nevykonaných událostí nemá na ukončení výpočtu vliv – pokud jsou v systému přítomny události, jejichž čas vykonání  $t_i$  je vyšší než hodnota `counter` a zároveň v systému nemohou být použita žádná běžná pravidla, je výpočet ukončen.

Pokud je hodnota `time` nastavena na `Inf`, běží simulace do doby, než dojde k přirozenému ukončení výpočtu (není možné vykonat žádné další pravidlo).

Aplikace pravidel (funkce `applyRules()`) je popsána algoritmem 5:

V každé membráně  $m$  z membránového systému  $ms$  (který je uložen jako globální

---

**Algoritmus 5** Výpočet funkce `applyRules()`

---

```
1: for all  $m \in ms@membranes$  do
2:   applicableRules = selectApplicableRules {vyber aplikovatelná pravidla}
3:   repeat
4:      $rulesApplied \ll -TRUE$ 
5:      $r = chooseRule(applicableRules)$  {vyber pravidlo}
6:     applyRule(r) {aplikuj pravidlo}
7:     applicableRules = selectApplicableRules {vyber aplikovatelná pravidla}
8:   until  $length(applicableRule) = 0$ 
9: end for
```

---

proměnná) jsou vybrána aplikovatelná pravidla. Jedno z nich je zvoleno na základě daných pravděpodobností a aplikováno. Proces je opakován tak dlouho, dokud jsou v membráně aplikovatelná pravidla.

Funkce, aplikující konkrétní pravidlo  $r$  se řídí následujícím algoritmem:

---

**Algoritmus 6** Výpočet funkce `applyRule(r)`

---

```
1:  $lbl = r@label$  {ve které membráně je pravidlo aplikováno}
2: odstraň objekty na levé straně pravidla  $r$  z membrány  $lbl$ 
3:  $pr = createPendingRule(r)$  {vytvoř visící pravidlo}
4:  $ms@pendingRules.push(pr)$  {přidej visící pravidlo do seznamu visících pravidel}
5: if  $r@synapseChange = TRUE$  then
6:    $ps = createPendingSynapse(r@synapse)$  {vytvoř visící synapsi}
7:    $ms@pendingSynapses.push(ps)$  {přidej visící synapsi do proměnné  $ms$ }
8: end if
```

---

Je nalezena membrána v níž je pravidlo vykonáno. Objekty na levé straně pravidla (včetně katalyzátoru) jsou odstraněny z membrány. Je vytvořeno visící pravidlo, které je přidáno do seznamu visících pravidel v globální proměnné  $ms$ . Pokud dochází během aplikace pravidla ke změně stavu synapse, je synapse přidána do seznamu visících synapsí v globální proměnné  $ms$ .

Z algoritmu je patrné, že během vykonání funkce `applyRule` vždy dojde pouze k odstranění objektů ze zdrojové membrány a nikdy ke vložení objektů do cílové membrány. Místo toho je vytvořeno visící pravidlo. Visící pravidlo obsahuje indikátor zdrojové membrány, cílové membrány, pravou stranu pravidla z něž je odvozeno, indikátor zda dochází ke změně stavu synapse, katalyzátor (z levé strany pravidla) a počet kroků, zbývajících do vykonání pravidla.

Počet kroků zbývajících do vykonání je při vytvoření o jedna nižší, než čas vykonání běžného pravidla, z něhož je visící pravidlo odvozeno. Běžné pravidlo  $[u]_i \xrightarrow{s' \rightarrow s''} [v]_j$  dá vzniknout visícímu pravidlu  $\xrightarrow{s' \rightarrow s''} [v]_j$ .

Důvod, proč je vytvořeno visící pravidlo i v případě, že je čas vykonání pravidla



roven jedné, je následující: výpočty v P systémech jsou paralelní. Pokud v jedné membráně může být aplikováno dvě nebo více pravidel, jsou aplikována současně. Není možné kompletně vykonat jedno pravidlo (odstranit objekty na levé straně a vložit objekty na pravé straně) a poté kompletně aplikovat další pravidlo.

Jeden způsob realizace výpočtu spočívá v určení úplné množiny pravidel, která budou v daném výpočetním kroku v dané membráně vykonána a pak je kompletně aplikovat jedno po druhém. Druhý způsob realizace výpočtu (zvolený zde) spočívá v postupném vykonávání pravidel tak, že je vybráno pravidlo, z membrány jsou odstraněny objekty na jeho levé straně a poté je znovu spočtena množina aplikovatelných pravidel. Poté, co jsou ve všech membránách aplikována všechna možná pravidla, je teprve přistoupeno k vložení objektů na pravé straně do membrán.

Vzhledem k tomu, že objekty jsou v rámci membránového systému přesunovány, není možné kompletně vykonat pravidla v některé z membrán a pak až přistoupit k aplikaci pravidel v další membráně. Úplné vykonání pravidel v membráně mění obsah ostatních membrán v systému. Výpočet musí být realizován tak, aby v průběhu jednoho výpočetního kroku bylo pracováno s konfigurací membránového systému, která existovala na počátku výpočetního kroku.

Je nutné nejprve odstranit objekty na levé straně všech aplikovaných pravidel ve všech membránách a až poté vložit všechny objekty na pravé straně těch pravidel, která byla aplikována.

Pokud by nebylo použito visících pravidel, bylo by nutné v průběhu výpočtu držet informaci o stavu membránového systému na počátku výpočetního kroku a informaci o aktuálním stavu membránového systému (kde již v průběhu výpočtu byla aplikována některá pravidla). Tyto dva objekty by musely být na konci každého výpočetního kroku synchronizovány.

Použití visících pravidel i v případě jednotkového času vykonání pravidel eliminuje potřebu pracovat se dvěma verzemi konfigurace P systému. Nejprve totiž dojde k odstranění všech objektů na levých stranách aplikovaných pravidel a až poté dojde ke vložení objektů na pravých stranách visících pravidel, která byla v aktuálním kroku dopočítána (jejich čas klesl na 0).

Tím je simulován paralelismus systému. Zde představený softwarový nástroj sekvenčně (jednu po druhé) prochází všechny membrány a v nich sekvenčně zpracovává pravidla v membránách.

Aplikace visících pravidel (funkce `applyPendingRule(pr)`) se řídí algoritmem:

---

**Algoritmus 7** Výpočet funkce `applyPendingRule(pr)`

---

```
1: rulesApplied << -TRUE
2: if pr@catalyst != NULL then
3:   vrať katalyzátor do zdrojové membrány
4:   pr@catalyst = NULL {odstraň katalyzátor}
5: end if
6: if pr@time != 0 then
7:   pr@time = pr@time -1
8:   RETURN SELF {vrať změněné visící pravidlo}
9: else
10:  vlož objekt na pravé straně pravidla do cílové membrány
11:  if ps@synapseChange = TRUE then
12:    změň stav synapse ps@synapse
13:    ms@pendingSynapses.pop(ps@synapse) {odstraň synapsi ze seznamu visících synapsí}
14:  end if
15:  RETURN NULL {pravidlo vykonáno, vrať NULL}
16: end if
```

---

Pokud pravidlo obsahuje katalyzátor, je vrácen do zdrojové membrány a odstraněn z visícího pravidla; tím pádem v dalším výpočetním kroku již není součástí visícího pravidla. To zajišťuje, že katalyzátor je vždy vrácen na konci prvního výpočetního kroku vykonání pravidla (pokud je součástí pravidla).

Pokud je čas *do vykonání* pravidla 1 nebo vyšší, je od času *do vykonání* odečtena jednotka a aplikace pravidla je ukončena.

Pokud je čas nulový, jsou do cílové membrány vloženy objekty na pravé straně pravidla. Pokud pravidlo popisuje změnu synapse, je změněn její stav a je odstraněna z množiny visících synapsí. Stav synapse je změněn až na konci posledního kroku vykonání pravidla, ačkoli měněná synapse je do seznamu visících synapsí vložena již v prvním kroku. To zaručuje, že nemůže být vykonáno víc pravidel, měnících stav jedné synapse, současně.

Při ukončení aplikace visícího pravidla funkce vrací hodnotu NULL, v opačném případě je vrácen pozměněný objekt nesoucí visící pravidlo.

V každém kroku vykonávání visícího pravidla je globální proměnná `rulesApplied` nastavena na TRUE. Tím si program udržuje informaci o tom, že probíhá neukončený výpočet. Pokud by tato hodnota nebyla nastavena, mohlo by dojít k situaci, že v systému nejsou žádná běžná pravidla, která by mohla být vykonána, ale existují visící pravidla, která jsou vykonávána. Systém by v takovém případě zastavil, protože na konci výpočetního kroku by nedostal informaci o to, že bylo vykonáno pravidlo.

Aplikace událostí (funkce `applyEvents(t)`) se řídí dle následujícího algoritmu:

---

**Algoritmus 8** Výpočet funkce `applyEvents(t)`

---

```
1: for all event ∈ ms@events do  
2:   if event@time = t then  
3:     if isApplicable(event@rule) then  
4:       rulesApplied << -TRUE  
5:       applyRule(event@rule)  
6:     end if  
7:   end if  
8: end for
```

---

Pro každou událost je zkontrolováno, zda jí přiřazený čas je shodný s aktuálním časovým krokem  $t$ . Pokud ano a pravidlo přiřazené události může být aplikováno, je aplikováno.

Pokud pravidlo nemůže být aplikováno v daném časovém kroku, výpočet pokračuje dále bez jeho vykonání.

Může se stát, že se systém dostane do stavu, kdy není možné vykonat žádná další pravidla a neexistují ani žádná právě vykonávaná visící pravidla, ale existují ještě nevykonané události, jejichž čas spuštění je vyšší než aktuální krok systému.

V takovém případě dojde k ukončení výpočtu.

Během jednoho výpočetního kroku systému dojde nejprve k aplikaci událostí, pokud je to možné. Pak jsou ve všech membránách aplikována běžná pravidla. Během aplikace běžných pravidel dojde k odstranění objektů na levé straně pravidla ze zdrojové membrány a k vytvoření visících pravidel. Po vykonání všech běžných pravidel dojde k vykonání visících pravidel. Během aplikace visících pravidel dojde ke vložení objektů na pravé straně pravidla, případně změně stavů synapsí.

Tato implementace událostí vede k tomu, že události „nesoutěží“ s dalšími pravidly přítomnými v membráně v níž má být událost vykonána. Pokud je v membráně přítomno pravidlo, popisující setrvání objektu a událost popisuje přemístění objektu, pak není mezi oběma možnostmi nedeterministicky vybíráno – dojde k přemístění objektu, protože aplikace událostí má prioritu.

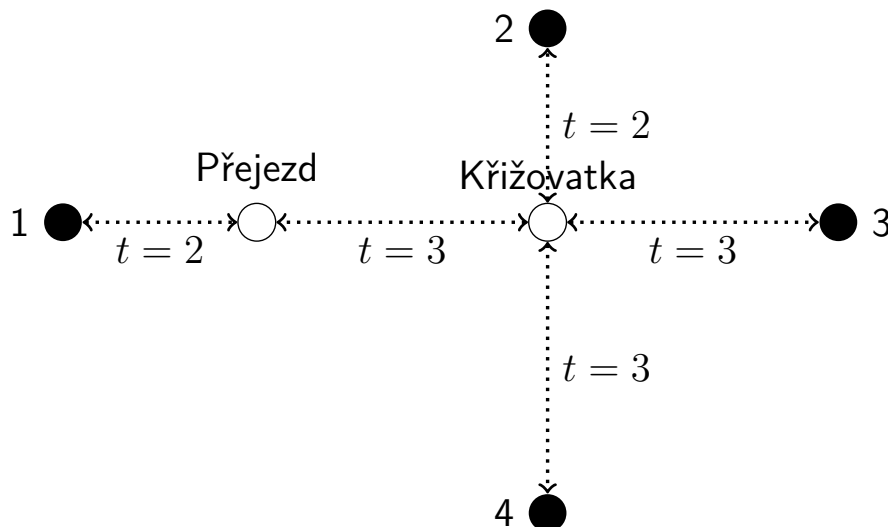
Alternativním způsobem implementace by bylo v kroku, v němž má být událost vykonána, vložit pravidlo asociované s událostí do membrány v níž má být vykonáno a na konci výpočetního kroku jej z membrány odstranit. Tento způsob by vedl k většímu nedeterminismu systému.

Každá událost je vykonána pouze jednou. Pokud událost popisuje přemístění objektu z membrány a v membráně jsou dva objekty, je přemístěn pouze jeden, zatímco

druhý zůstává a mohou na něj být aplikována běžná pravidla. Vzhledem k tomu, že události mají reprezentovat jednorázové, nenadálé jevy, není je limitující vlastnost. V případě, že by událost měla být použita například k přemístění všech objektů z membrány (bez znalosti jejich počtu), byla by vhodnější implementace, kdy je pravidlo události vloženo přímo do membrány a na konci časového kroku opět odebráno. Tak by totiž mohlo dojít k několikanásobnému vykonání pravidla (ovšem bez priority, pravidlo události by bylo běžným pravidlem). Priorita vykonání události nad ostatními pravidly lze reprezentovat jako nekonečná pravděpodobnost (v jazyce  $\mathbf{R}$  konstanta  $\text{Inf}$ , která je nejvyšším možným číslem).

#### 4.4 Simulace: vliv kapacity křižovatky na rozpuštění fronty

Simulace uvažuje systém, vyobrazený na obrázku 4: Existují 4 body (brány), sloužící jako bílé a černé díry, křižovatka a železniční přejezd. Brána je ve skutečnosti dvěma membránami, z nichž jedna slouží jako černá díra a jedna jako bílá díra. Tyto dvě membrány nejsou spojeny synapsí, existují pouze synapse vedoucí z bílé díry do sousední membrány a ze sousední membrány do černé díry. Tři brány jsou spojeny přímo s křižovatkou. Čtvrtá brána je spojena s přejezdem, který je dále spojen s křižovatkou.



Obrázek 4: Schéma membránové sítě pro simulaci rozpuštění fronty

Křižovatka má danou kapacitu  $k$ , v bílých děrách jsou generovány vozidla každou časovou jednotku. Přejezd je reprezentován jako křižovatka bez signalizace s neomezenou kapacitou (vozidla mohou membránu opouštět, pokud existuje otevřená synapse, přítomnost katalyzátoru není nutná). Čas nezbytný k překonání synapsí je zobrazen na obrázku 4 (označen jako  $t$ ). Membrána označená jako *Přejezd* je křižovatkou, která neobsahuje signalizaci a je synapsemi spojená jen se dvěma dalšími membrá-

nami. Všechny synapse vedoucí do a z přejezdu jsou na počátku simulace otevřeny. Průjezd přejezdem trvá dvě časové jednotky (v prvním kroku je do membrány přiveden objekt  $veh$ , který se změní na objekt  $veh(in_j)$  a ve druhém kroku membránu opouští). Systém se tedy chová stejně, jako kdyby byly membrána 1 a *Křižovatka* spojeny přímo synapsí.

V systému jsou definovány události, které simulují průjezd vlaku přes přejezd. Události jsou definovány jako pravidla popsaná rovnicemi 29 a 30, která nejprve uzavřou (v čase  $t = 23$ ) a pak otevřou (v čase  $t = 30$ ) synapse vedoucí z přejezdu do sousedních membrán (křižovatka a brána 1).

Membrána *Přejezd* je v systému modelována jako křižovatka. To však není správná reprezentace s ohledem na realističnost výběru cest vozidel. Jelikož je předpokládána stejná pravděpodobnost výběru pro všechny synapse, vozidla, která se ocitnou v membráně *Přejezd* budou s 50% pravděpodobností přemístěny do membrány 1 a s 50% pravděpodobností do membrány *Křižovatka*.

Pokud například v jednom kroku z membrány *Křižovatka* přijde 20 vozidel a z membrány 1 přijde 5 vozidel, očekávané chování systému je takové, že 20 vozidel bude pokračovat do membrány 1 a 5 vozidel bude pokračovat do membrány *Křižovatka*. Ve skutečnosti však dojde k tomu, že (v průměru) každou ze synapsí bude přemístěna polovina objektů. Do membrány *Křižovatka* bude směřovat 12,5 namísto 5 objektů.

K membráně *Přejezd* tedy nelze přistupovat jako ke křižovatce, ale je nutné změnit množinu pravidel tak, aby byl zajištěn jednosměrný průchod vozidel. Namísto pravidla

$$\left[ veh(in_j), k \mid syn_{(i,j)}^+ \right]_i \xrightarrow{t} \left[ veh \right]_j \quad (33)$$

je použito pravidlo

$$\left[ veh(in_j), k \mid syn_{(i,j)}^+ \right]_i \xrightarrow{t} \left[ veh(in_k) \right]_j. \quad (34)$$

Namísto obecného vozidla je v membráně *Přejezd* vytvořeno vozidlo, které již má určený cíl. Takto definovaný systém modeluje průjezd železničním přejezdem korektně. Zároveň dojde ke zkrácení cesty mezi membránami 1 a *Křižovatka* o jeden krok (nedochází k rozhodování, kterou membránou bude transport realizován) a je tedy nutné navýšit čas  $t$  synapse (1, *Přejezd*) o jednotku.

Membrána *Křižovatka* obsahuje pravidla, měnící stav synapsí, které z membrány vedou do sousedních membrán. Na počátku výpočtu jsou synapse

(*Křižovatka, Přejezd*) a (*Křižovatka, 3*) otevřeny a synapse (*Křižovatka, 1*) a (*Křižovatka, 2*) uzavřeny. Stav synapsí se mění po dvou krocích. V každém okamžiku jsou otevřeny dvě synapse vedoucí z *Křižovatky*, zatímco dvě jsou uzavřeny.

Intenzita dopravy je na všech komunikacích totožná, tudíž pravděpodobnost výběru pravidla je stejná pro všechny cesty (pro všechny membrány, dostupné z membrány *Křižovatka*).

Jsou porovnány 4 varianty nastavení systému:

Přejezd vlaku	Kapacita křižovatky
Ne	15
Ano	15
Ano	20
Ano	25

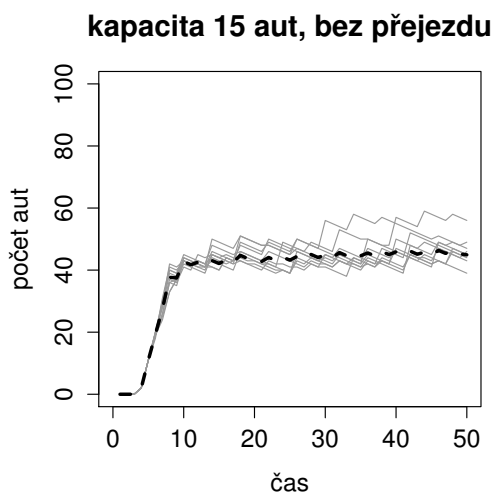
Tabulka 1: Nastavení parametrů simulace rozpuštění fronty

První dvě porovnávané situace se liší tím, zda přes přejezd přejede vlak. Pokud ano, dojde po dobu 6 kroků k uzavření synapsí vedoucích z membrány *Přejezd*. Poté, co jsou synapse znovu otevřeny již zůstanou otevřené.

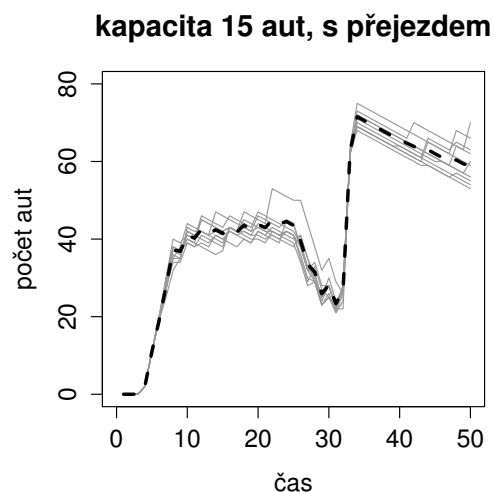
Druhým parametrem, který se mezi jednotlivými simulacemi liší, je kapacita křižovatky, vyjádřená počtem vozidel, která mohou křižovatkou projít za jednu časovou jednotku. V membránách 1 a 3 je každou časovou jednotku generováno 5 vozidel, v membránách 2 a 4 jsou generována 2 vozidla. Počet generovaných vozidel je konstantní po celou dobu výpočtu. Zde prezentované výsledky ukazují simulaci po dobu 50 časových jednotek. Pro každou ze čtyř porovnávaných situací je provedeno 50 simulací. Na obrázcích 5 až 8 jsou zaznačeny počty vozidel v membráně *Křižovatka* v jednotlivých simulacích (světle šedá barva) a jejich průměr (černá čárkovaná linie). Zobrazené počty ukazují všechny objekty reprezentující vozidla, tedy jak objekty *veh*, tak *veh(in<sub>j</sub>)*. Katalyzátory nejsou v počtu zahrnuti.

Ačkoli počet vozidel, která jsou v každém kroku generována je konstantní a stav signalizace je měněn ve stále stejných časových krocích, je systém nedeterministický. Pravidlo 27 umožňuje vozidlu, které přijde do membrány, aby se v dalším kroku vrátilo do membrány, z níž přišlo. Jinak řečeno, mezi jednotlivými vozidly se nerozlišuje a jsou-li v membráně přítomny jako objekt *veh*, není již možné zjistit, ze které membrány do současné membrány přišly a sledovat historii jejich pohybu v systému. Náhodný výběr membrán, do nichž budou objekty přemísťovány (pravděpodobnosti výběru všech synapsí jsou totožné) přináší do systému nedeterminismus.

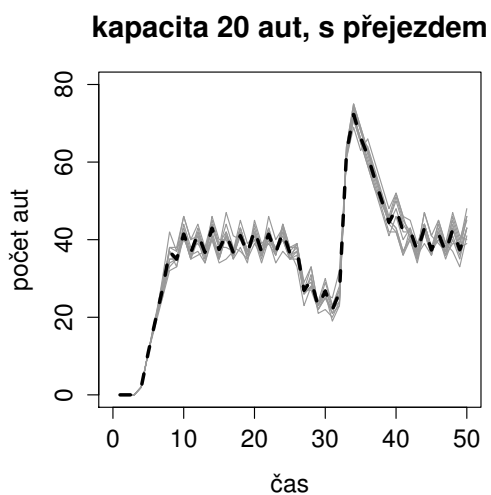
Z porovnání obrázků 5 a 6 je patrné, že pokud dojde k přejezdu vlaku – dočasněho



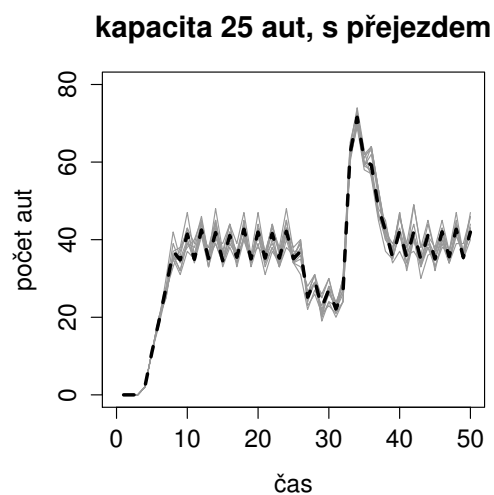
Obrázek 5: Kapacita 15 vozidel za časovou jednotku, bez přejezdu vlaku



Obrázek 6: Kapacita 15 vozidel za časovou jednotku, s přejezdem vlaku



Obrázek 7: Kapacita 20 vozidel za časovou jednotku, s přejezdem vlaku



Obrázek 8: Kapacita 25 vozidel za časovou jednotku, s přejezdem vlaku

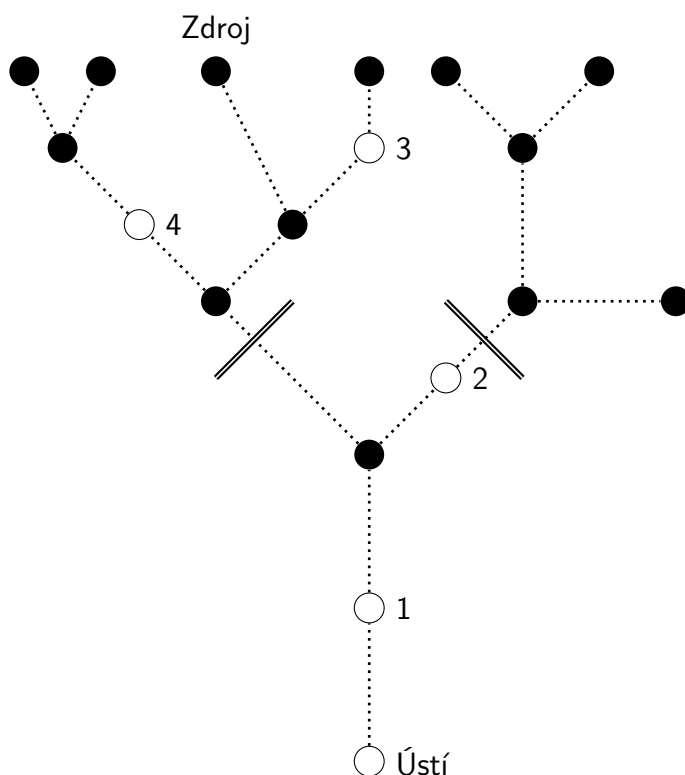
uzavření synapsí vedoucích z membrány *Přejezd* – je nejprve snížen počet přijíždějících vozidel, který je následně kompenzován náhlým přísunem téměř 50 vozidel, která jsou v několika časových krocích z křižovatky odvedena. Rychlost mizení fronty závisí na kapacitě křižovatky. V případě kapacity 15 vozidel dochází k rozpouštění fronty velmi pomalu, zatímco při kapacitě 20 vozidel zmizí fronta za 10 časových jednotek. Se zvyšující se kapacitou (25 vozidel) již k výraznému urychlení odbavení vozidel v křižovatce nedochází. Po zmizení fronty osciluje počet vozidel v křižovatce okolo 40 (započítána jsou všechna vozidla).

Provedená simulace nepřináší výsledky, které by byly v některém ohledu překvapivé, nicméně demonstruje schopnost navrženého modelu P systémů realisticky mode-

lovat dopravní proud a to včetně situací, které nelze v rámci některých modelovacích přístupů jednoduše řešit. Použití *událostí* je rozšířením P systémů, které lze jednoduše použít pro modelování náhodných, nepravidelných nebo málo četných událostí.

#### 4.5 Simulace: šíření invazivního druhu říční sítí

Druhá simulace demonstruje použití *transportních P systémů* pro modelování šíření invazivních druhů.



Obrázek 9: Navržená síť pro simulace šíření invazivních druhů a šíření znečištění

Modelovaný systém je navržen jako říční síť obsahující rybníky. Rybníky jsou na obrázku 9 označeny čísly 1 až 4. Membrána *Ústí* je koncovým bodem systému.

Černě označené membrány reprezentují soutoky a prameny. V *Ústí* je na počátku výpočtu 250 objektů  $a$ , reprezentujících organismus. Pomocí synapsí, které jsou mezi každými sousedícími membrány vždy v obou směrech, dochází k pohybu organismu sítí. Všechny membrány obsahují pravidla pro transport sítí v obou směrech i pravidlo, popisující setrvání. Pravidla mají asociované pravděpodobnosti, přičemž pravděpodobnost pohybu po proudu a proti proudu se liší.

Membrány dále obsahují katalyzátory, které reprezentují potravní zdroje. Pravidlo popisující setrvání v membráně je podmíněno přítomností potravy – katalyzátoru.



Signalizace je použita k simulaci zámru řeky. V obrázku 9 reprezentuje dvojitá čára kolmá k synapsi hranici zámru řeky. Synapsím nad hranicí zámru je periodicky měněn stav, přičemž stav  $s^-$  má význam zamrzlé řeky. Pokud je synapse v uzavřeném (zamrzlém) stavu, není ji možné použít k transportu objektů. Synapse pod hranicí zámru zůstávají v otevřeném stavu  $s^+$  po celou dobu výpočtu.

Parametry systému shrnuje tabulka:

Parametr	Hodnota
Počet organismů v <i>Ústí</i> (v počátku výpočtu)	250
Množství potravy v <i>Ústí</i>	500
Množství potravy v <i>Rybnících</i>	50
Množství potravy v <i>soutocích a pramenech</i>	5
Pravděpodobnost setrvání v membráně	10 / 10 / 100
Pravděpodobnost opuštění membrány	5 / 5 / 50
Pravděpodobnost úhynu	0 / 1 / 1
Čas pro překonání synapse	2
Čas po který je synapse otevřená (splavná)	9
Čas po který je synapse uzavřená (zamrzlá)	3

Tabulka 2: Nastavení parametrů simulace šíření invazivního druhu

Jsou porovnány tři nastavení systému, lišící se v pravděpodobnosti úhynu. Pravděpodobnost setrvání v membráně je vždy dvojnásobná oproti pravděpodobnosti opuštění membrány. V jednom případě je úhyn vyloučen ( $p_{\text{úhyn}} = 0$ ), v jednom případě je jeho pravděpodobnost desetinná oproti pravděpodobnosti setrvání ( $p_{\text{úhyn}} = 1$ ,  $p_{\text{setrvání}} = 10$ ) a v poslední variantě nastavení je setinová ( $p_{\text{úhyn}} = 1$ ,  $p_{\text{setrvání}} = 100$ ) v porovnání s pravděpodobností setrvání.

Množina pravidel systému je:

- **Setrvání v membráně**

$$\left[ a \ k \right]_i \xrightarrow{t=1,p} \left[ a \right]_i, \quad (35)$$

Toto pravidlo se nachází ve všech membránách a popisuje situaci, kdy živočich setrvává ve stávající membráně. K vykonání pravidla je nutné, aby byl v membráně přítomen katalyzátor  $k$ , reprezentující potravu. S pravidlem je asociována pravděpodobnost  $p$ .

- **Opuštění membrány**

$$\left[ a \mid \text{syn}_{(i,j)}^+ \right]_i \xrightarrow{t,p} \left[ a \right]_j, \quad (36)$$

Toto pravidlo se nachází ve všech membránách a přesune organismus do některé ze sousedních membrán, pokud existuje otevřená synapse. Vykonání trvá  $t$  časových jednotek, s pravidlem je asociována pravděpodobnost  $p$ .

- **Úhyn**

$$\left[ a \right]_i \xrightarrow{t=1,p} \left[ \right]_i, \quad (37)$$

Toto pravidlo se nachází ve všech membránách a odstraní jednu instanci objektu  $a$  z membrány, čímž je simulován úhyn organismu.

- **Zamrznutí a rozmrznutí řeky**

$$\left[ \lambda \mid syn_{(i,j)}^+ \right]_i \xrightarrow[s^+ \rightarrow s^-]{t} \left[ \lambda \right]_i, \quad (38)$$

a

$$\left[ \lambda \mid syn_{(i,j)}^- \right]_i \xrightarrow[s^- \rightarrow s^+]{t} \left[ \lambda \right]_i, \quad (39)$$

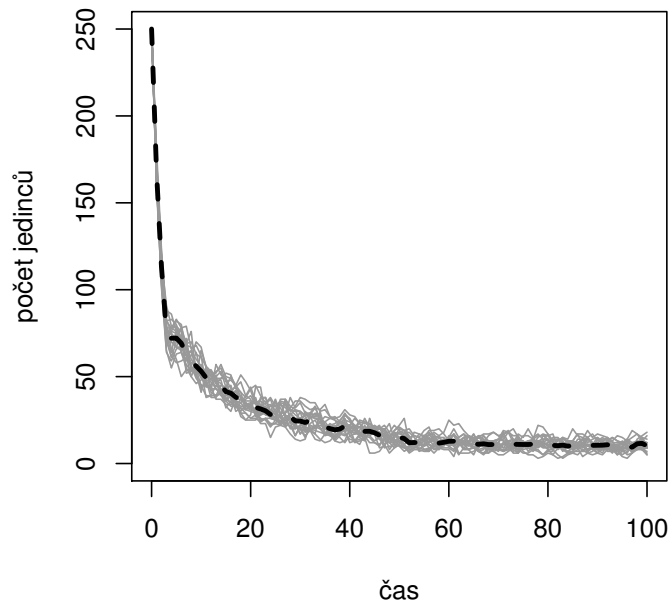
Tato pravidla se nachází ve všech membránách, které jsou „nad hranicí zámrazu“ nebo s těmito membránami přímo sousedí. Délka trvání zamrznutí a rozmrznutí řeky se liší. Symbol  $\lambda$  označuje prázdnou množinu – k vykonání pravidla není nutné, aby se v membráně nacházel jakýkoli objekt.

## Výsledky

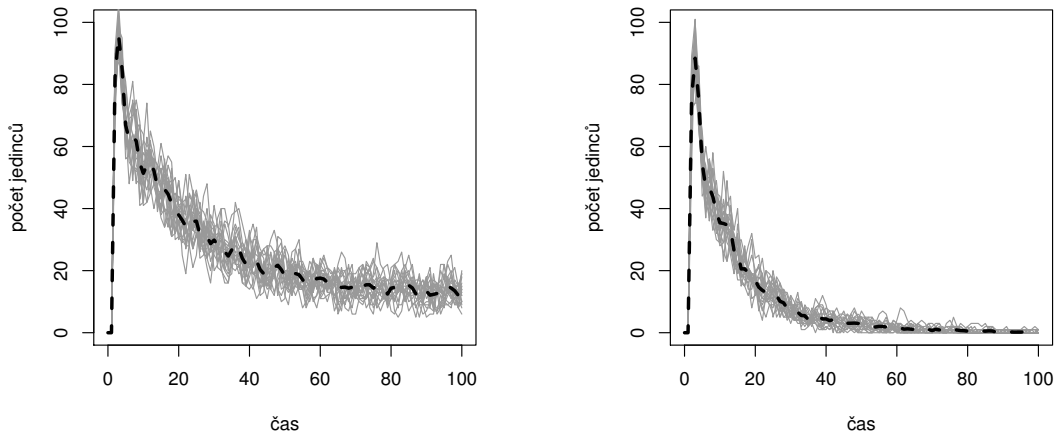
Všechna tři nastavení systému byla modelována 100 krát po dobu 100 kroků.

V membráně *Ústí* při všech nastaveních systému dojde k rychlému poklesu počtu objektů (katalyzátory nejsou počítány). V případě, že je modelován úhyn ( $p_{\text{úhyn}} \neq 0$ ), se počty limitně blíží nule. V případě, že není modelován, se na konci simulace (po 100 krocích) v membráně nachází okolo 20 jedinců (obr. 10), ačkoli je v membráně přítomno 500 katalyzátorů. Pokles počtu objektů je hyperbolický.

V membráně označené 1 (reprezentující rybník) dojde k prudkému nárůstu počtu jedinců na hodnotu cca 100, po kterém opět následuje pokles, jehož tvar se blíží hyperbole. Snižování počtu je pomalejší než v membráně *Ústí*, přičemž pokud je počítáno s úhynem, dojde během 100 kroků simulace k vymření a v membráně nejsou přítomny žádné objekty (obr. 12). Pokud s úhynem není počítáno, nachází se po 100 krocích v membráně v průměru 15 objektů, ačkoli membrána obsahuje 50 katalyzátorů, sloužících jako potravní zdroje (obr. 11).



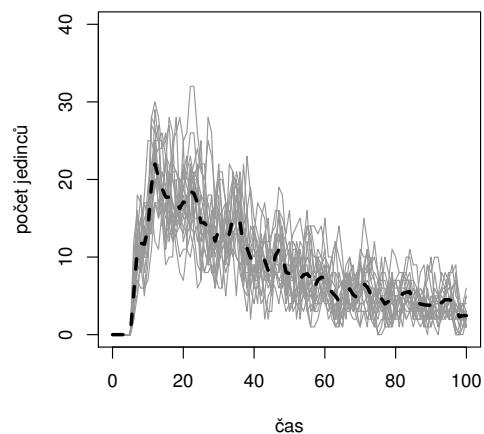
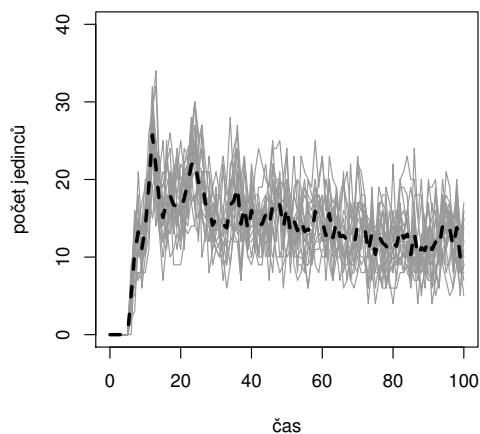
Obrázek 10: Počet jedinců v *Ústí*,  $p_{úhyn} = 0$



Obrázek 11: Počet jedinců v membráně 1,  $p_{úhyn} = 0$   
 Obrázek 12: Počet jedinců v membráně 1,  $p_{úhyn} = 1, p_{setrvání} = 10$

Membrána 2 (reprezentující rybník) je od *Ústí* vzdálenější (cesta přes 3 synapse) a i když vývoj početnosti vykazuje podobný trend jako u membrán *Ústí* a 1, je maximální počet jedinců v systému nižší a pokles počtu pomalejší. V membráně 1 nedochází k zamrznání, v membráně 2 dochází k zamrznání jedné synapse ze dvou, které do membrány 2 směřují. Zamrznání se projevuje cyklickými výkyvy, korespondujícími s periodami zámru. U všech zatím zkoumaných membrán byl pozorován stejný trend

prudkého nárůstu počtu jedinců a následného poklesu, který má tvar hyperboly. Čím dále od *Ústí* se membrána nachází, tím je maximum počtu objektů v membráně nižší a tím je pokles pozvolnější. Pokud je počítáno s úhynem, dochází vždy k vymření v dané membráně.

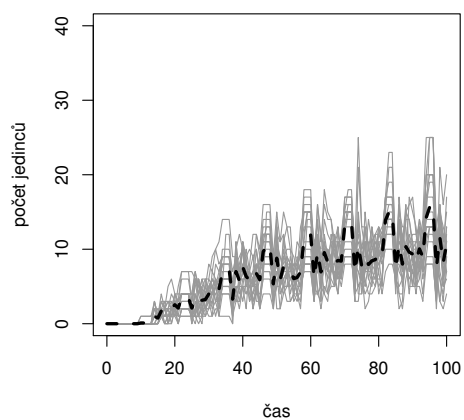


Obrázek 13: Počet jedinců v membráně 2,  $p_{\text{úhyn}} = 0$   
 Obrázek 14: Počet jedinců v membráně 2,  $p_{\text{úhyn}} = 1, p_{\text{setrvání}} = 100$

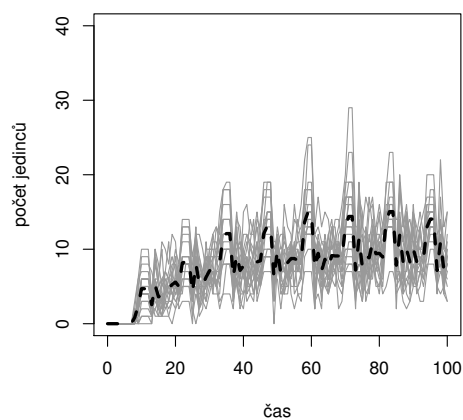
Membrány 3 a 4, které rovněž reprezentují rybníky, ale nacházející se v části říční sítě, která zamrzá, vykazují odlišný trend v počtu jedinců (obr. 15 a) až f).

V žádné z membrán nedochází k prudkému nárůstu počtu jedinců, naopak nárůst je pozvolný a bez výrazného vrcholu. V případě, že není počítáno s úhynem, se počet jedinců v membráně ustálí okolo hodnoty 10 jedinců, ačkoli membrány obsahují potravní zdroje pro 50 jedinců.

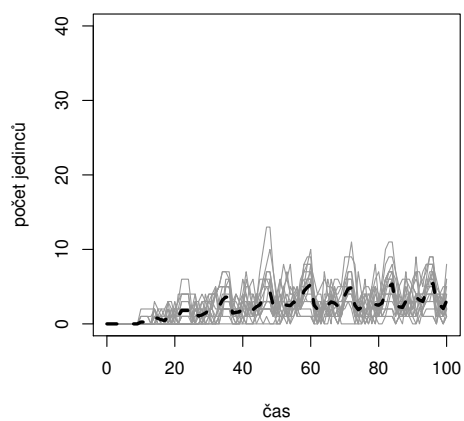
Pokud systém obsahuje pravidla pro úhyn, dojde ke snižování počtu jedinců až k nule, přičemž čím větší je relativní pravděpodobnost úhynu, tím menších maxim počet jedinců dosahuje. Maximální počet jedinců se rovněž snižuje se vzdáleností od *Ústí*.



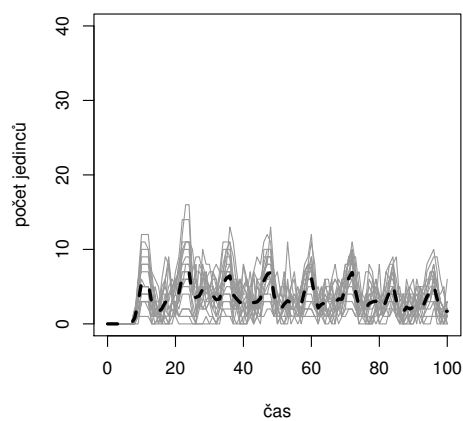
(a) Počet jedinců v 3,  $p_{úhyn} = 0$



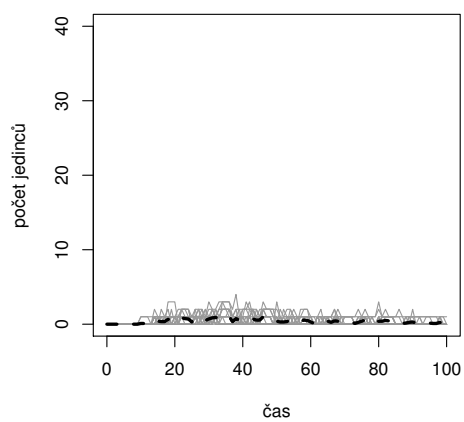
(b) Počet jedinců v 4,  $p_{úhyn} = 0$



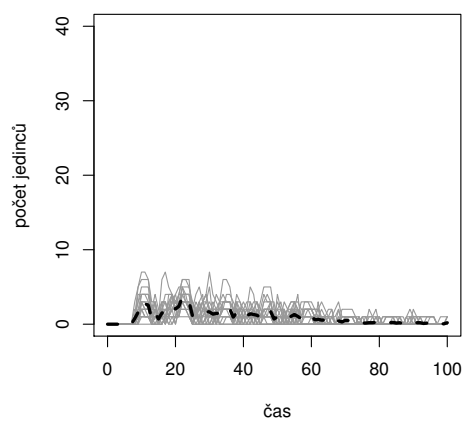
(c) Počet jedinců v 3,  
 $p_{úhyn} = 1, p_{setrvání} = 100$



(d) Počet jedinců v 4,  
 $p_{úhyn} = 1, p_{setrvání} = 100$



(e) Počet jedinců v 3,  
 $p_{úhyn} = 1, p_{setrvání} = 10$



(f) Počet jedinců v 4,  
 $p_{úhyn} = 1, p_{setrvání} = 10$

Obrázek 15: Vývoj početnosti jedinců v membránách 3 a 4

Výsledky ukazují, že čím vzdálenější je membrána od *Ústí*, které představuje zdroj objektů, tím pomalejší je nárůst počtu jedinců, tím nižší je maximum, kterého je v membráně dosaženo a tím pomalejší je pokles po dosažení maxima.

V membránách, do nichž vedou zamrzající úseky říční sítě jsou pozorovány cyklické výkyvy, korespondující s periodou zamrzání, v membránách v nezamrzající části říční sítě tyto výkyvy patrné nejsou.

Ačkoli membrány reprezentující rybníky obsahují katalyzátory, umožňující setrvání 50 jedinců, hodnota na níž se počet jedinců ustálí, je nižší. Namísto koncentrace do rybníků je pozorována disperze, která směřuje k rovnoměrnému rozmístění objektů ve všech membránách bez ohledu na potravní zdroje. V případě, že je v systému přítomno 250 jedinců a systém obsahuje 18 membrán, vychází v průměru 14 jedinců na membránu. Při simulaci 1000 kroků dojde k rozptylu a všechny membrány obsahují v průměru okolo 14 jedinců, přičemž počty v membránách nezůstávají konstantní, ale kolísají v rozmezí cca 3 až 20 objektů v každé membráně. To platí i pro membrány reprezentující prameny a soutoky (grafy zde nejsou uvedeny).

V případě, že existují pravidla popisující úhyn, dochází ve všech membránách k vymření. Čím větší je relativní pravděpodobnost úhynu v porovnání s pravděpodobností setrvání, tím menších maxim dosahuje počet objektů v jednotlivých membránách, přičemž čím dále se membrána nachází od zdroje (*Ústí*), tím je tento efekt výraznější.

*Transportní P systémy* neumožňují vytvořit pravidla popisující růst, jediným zdrojem objektů v systému je vnější prostředí, což limituje použití pro věrohodnější modelování populací, v principu však není problém model rozšířit tak, aby na levé i pravé straně pravidla mohlo být více než jen jeden objekt (a tudíž model byl schopen reprezentovat růst populací). Takový model není v této práci podrobněji zkoumán, neboť primárním cílem práce je zkoumat transport v P systémech a nikoli vývoj populací.

Setrvání v membráně popisuje pravidlo, které obsahuje katalyzátor, reprezentující potravu, zatímco pravidlo popisující odchod z membrány katalyzátor neobsahuje. Namísto katalyzátoru  $k$  by zde bylo možné použít inhibitor  $\neg k$ , jehož význam by byl „pokud není přítomen objekt  $k$ , vykonej pravidlo“. Tento způsob zápisu lépe popisuje pohyb živočichů v závislosti na dostatku potravních zdrojů než pravděpodobnosti asociované s pravidly, vyžadoval by však úpravy navrženého modelu, který s inhibitory nepracuje. Potenciálním problémem při použití inhibitorů je přílišný determinismus systému; pokud existují pouze dvě pravidla, z nichž jedno popisuje setrvání a vyžaduje přítomnost katalyzátoru a druhé popisuje opuštění membrány a vyžaduje jeho

absenci, pak jsou pravidla vykonávána přísně deterministicky v závislosti na počtu katalyzátorů v membráně a samovolné opuštění membrány navzdory přítomnosti potravních zdrojů není možné modelovat.

Z výše řečeného vyplývá, že model není zcela vhodný k modelování populačního růstu a/nebo metapopulací (prostorově oddělených populací), nicméně může sloužit jako základ, který je rozšířen pro účely biologického modelování. Jako nezbytné se jeví zahrnout do modelu pravidla umožňující reprezentaci růstu populace a detailnější popis mechanismu disperze populace. Dynamický výpočet pravděpodobnosti opuštění membrány na základě aktuální konfigurace membrány je jednou z možností, použitou například v práci (Pérez-Jiménez a Romero-Campero, 2006); druhou možností je nepracovat s pravděpodobnostmi, ale popsat disperzi pravidly, pracujícími s katalyzátory, promotory/inhibitory a dalšími prvky, které omezují nedeterminismus systému.

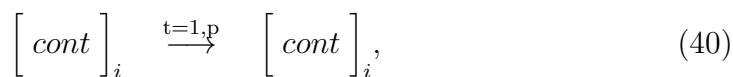
#### 4.6 Simulace: šíření znečištění říční sítí

Je zkoumána hypotetická situace, kdy dojde k bodovému znečištění vodního toku. Pro simulaci je použita membránová struktura, která byla použita v předchozí simulaci. Na obrázku 9 je jako *Zdroj* označena membrána, v níž dojde ke znečištění. Znečištění je reprezentováno pomocí objektu *cont*. Je zkoumána doba, kterou trvá, než znečištění opustí systém (než budou všechny objekty *cont* v membráně *Ústí*) a je sledováno, jakých koncentrací znečištění nabývá v jednotlivých membránách během výpočtu.

Navržený model nepracuje se signalizací (všechny membrány jsou otevřené) ani s katalyzátory.

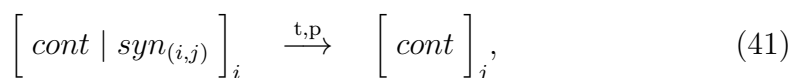
Model obsahuje následující množinu pravidel:

- **Setrvání v membráně**



Toto pravidlo se nachází ve všech membránách s výjimkou *Ústí* a popisuje situaci, kdy znečištění setrvává v aktuální membráně (ať již se jedná o rybník, soutok nebo pramen). S pravidlem je asociována pravděpodobnost  $p$ .

- **Opuštění membrány**



Toto pravidlo se nachází ve všech membránách kromě *Ústí* a přesune objekt *cont* do některé ze sousedních membrán, pokud mezi dvojicí membrán existuje synapse. Vykonání trvá  $t$  časových jednotek, s pravidlem je asociována pravděpodobnost  $p$ . Je vyžadována pouze existence synapse, na jejím stavu nezáleží (model předpokládá, že jsou všechny synapse po celou dobu výpočtu otevřené).

Membrána *Ústí* neobsahuje žádná pravidla – pokud se objekt *cont* dostane do této membrány, již zde zůstane. Oproti výše představeným modelům dochází k výpočtu tak, jak je definovaný v klasické variantě P systémů. V okamžiku, kdy jsou všechny objekty *cont* přítomny v membráně *Ústí*, není možné použít již žádné další pravidlo a výpočet končí. Předchozí simulace sledovaly stav systému v jednotlivých časových krocích, zde je sledován počet kroků, které je nutné provést než dojde k úspěšnému ukončení výpočtu.

Z popisu modelu je zřejmé, že je zjednodušením *transportních P systémů se signalizací* a stejný systém by bylo možné simulovat pomocí jiných variant P systémů, například generalizovaných komunikačních P systémů (Verlan et al., 2008).

Model obsahuje pouze tři parametry - pravděpodobnost setrvání v membráně ( $p_{here}$ ), pravděpodobnost transportu po proudu ( $p_{down}$ ) a pravděpodobnost transportu proti proudu ( $p_{up}$ ). Jsou sledovány následující charakteristiky modelu:

- Počet časových kroků, než budou všechny objekty *cont* přemístěny do membrány *Ústí* (počet časových kroků do ukončení výpočtu).
- Informace, zda během výpočtu dojde k přemístění objektu *cont* do předem určených membrán. Můžeme například předpokládat, že rybníky obsahují křehký ekosystém, který v přítomnosti polutantu zkolabuje a sledovat s jakou pravděpodobností dojde při dané konfiguraci ke znečištění rybníků.

Jsou simulovány tři scénáře, přičemž čas transportu mezi dvěma synapsi je vždy 3 časové jednotky. Pro každý scénář je provedeno 100 simulací, jejichž výsledky jsou popsány níže.

**První scénář** předpokládá vysokou pravděpodobnost transportu po proudu ( $p_{down} = 10$ ), střední pravděpodobnost setrvání ( $p_{here} = 5$ ) a nízkou pravděpodobnost transportu proti proudu ( $p_{up} = 1$ ).

U všech rybníků dojde ve více než polovině případů ke kontaminaci; doba, po kterou jsou znečišťující látky přítomny v systému se pohybuje v průměru okolo 50 časových kroků. Nejkratší cesta z membrány *Zdroj* do *Ústí* trvá 22 časových jednotek.



<b>Rybník</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
Kontaminace (% případů)	100	59	87	71
<b>Doba výpočtu</b>	minimum 35	medián 48	průměr 50	maximum 82

Tabulka 3: Výsledky 100 simulací,  $p_{here} = 5, p_{up} = 1, p_{down} = 10$

Při tomto nastavení trvá „vyčištění“ systému od znečišťujících látek v průměru více než dvojnásobek nejkratší možné doby.

**Druhý scénář** předpokládá nízkou pravděpodobnost transportu po proudu ( $p_{down} = 2$ ), i setrvání ( $p_{here} = 2$ ) a nízkou pravděpodobnost transportu proti proudu ( $p_{down} = 1$ ).

<b>Rybník</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
Kontaminace (% případů)	100	94	100	100
<b>Doba výpočtu</b>	minimum 181	medián 286	průměr 286	maximum 478

Tabulka 4: Výsledky 100 simulací,  $p_{here} = 2, p_{up} = 1, p_{down} = 2$

Při tomto nastavení, které předpokládá malý rozdíl mezi pravděpodobnostmi trvá v průměru téměř 300 časových jednotek, než dojde k ukončení výpočtu a odstranění všech znečišťujících látek. Zároveň téměř při všech simulacích došlo ke kontaminaci všech rybníků (jediný rybník 2 nebyl kontaminován ve všech bžích simulátoru).

**Třetí scénář** předpokládá velmi vysokou pravděpodobnost transportu po proudu ( $p_{down} = 100$ ), střední pravděpodobnost setrvání ( $p_{here} = 10$ ) a nízkou pravděpodobnost transportu proti proudu ( $p_{up} = 1$ ).

<b>Rybník</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
Kontaminace (% případů)	100	0	0	0
<b>Doba výpočtu</b>	minimum 22	medián 23	průměr 24	maximum 32

Tabulka 5: Výsledky 100 simulací,  $p_{here} = 10, p_{up} = 1, p_{down} = 100$

Ve všech bžích simulace došlo k odstanění kontaminujících látek ze systému za méně než jeden a půl násobek nejkratší možné doby. Polovina z kontaminujících látek byla odstraněna za dobu jen o jednu jednotku delší, než je nejkratší možná doba. Jediný rybník, kde došlo ke kontaminaci, je rybník 1, který se však nachází na cestě ze *Zdroje* znečištění do *Ústí* a kontaminován být musí. U žádného z ostatních rybníků nedošlo v žádné ze simulací ke kontaminaci.

Transport látek říčním proudem je proces, který je ze své podstaty mechanický. Přenášený materiál nečiní rozhodnutí a nemůže aktivně ovlivnit svou cestu sítí. Z tohoto pohledu se jedná o ideální proces pro modelování v P systémech (zde představená simulace uvažovala i možnost transportu látek proti proudu, ovšem čistě kvůli demonstraci použití modelu). Kromě šíření znečištění je možné modelovat například sedimentaci či erozi, tání nebo naopak vyčerpávání vodních zásob. Detailní modelování těchto jevů sice P systémy neumožňují (pro všechny výše zmíněné jevy existují podrobné fyzikální modely), nicméně na úrovni celých povodí nebo i vyšších geografických celků můžou P systémy poskytovat přiměřenou míru generalizace a zároveň být dostatečně robustním a parametrizovatelným modelem.

## 4.7 Simulace: jednoduchá náhodná procházka

Náhodná procházka je jednoduchý stochastický proces, který formalizuje intuitivní proces provádění náhodných kroků (Pearson, 1905; Feller, 1968) v určitém matematickém prostoru, například množině celých čísel. Jednoduchá náhodná procházka je náhodná procházka v jednom rozměru, tedy na přímce.

Teorii náhodných procházek lze aplikovat v oborech finančnictví (Steele, 2001), biologie (Codling et al., 2008) nebo fyziky (zejména kinetiky kapalin a plynů) (Blumen et al., 1986) a přes svou jednoduchost představuje matematický model s rozsáhlým aplikačním potenciálem.

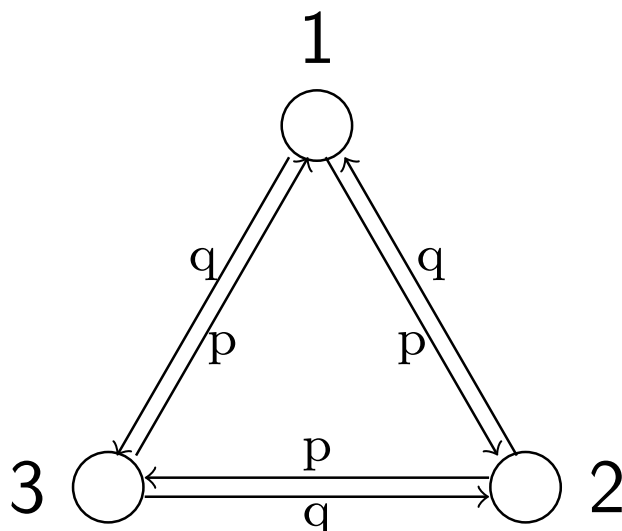
*Transportní P systémy* mohou sloužit jako jednoduchý simulátor náhodných procházek. Obrázek 16 zobrazuje navrženou membránovou strukturu. Skládá se ze tří membrán (1,2,3), které jsou mezi sebou spojeny pomocí synapsí. Čas cesty po synapsi je vždy roven jedné. Každá synapse má přiřazenu hodnotu pravděpodobnosti použití synapse ( $p, q$ ). Obecně se mohou hodnoty  $p$  a  $q$  lišit, pro náhodnou procházku však platí  $p = q = \frac{1}{2}$ .

Všechny membrány obsahují pravidla

$$\left[ veh \mid syn_{(i,j)} \right]_i \xrightarrow{t=1,p} \left[ veh \right]_j, \quad (42)$$

která popisují přemístění objektu  $veh$  z membrány  $i$  do membrány  $j$ . Podmínkou je existence synapse  $(i, j)$ , která obě membrány spojuje.

Žádná další pravidla nejsou v žádné z membrán přítomna. Každá membrána obsahuje 2 pravidla, popisující transport objektu  $veh$  z membrány jedním nebo druhým směrem.



Obrázek 16: Membránová struktura pro simulaci jednoduché náhodné procházky

Na počátku výpočtu je jeden objekt *veh* přítomen v membráně 1, ostatní membrány jsou prázdné.

Během výpočtu dochází k přemísťování objektu *veh* membránovou strukturou, přičemž nedojde k zastavení systému. Proto je nutné stanovit počet  $t$  časových kroků, po které bude systém počítat.

Výsledkem výpočtu je seznam konfigurací v každém kroku simulace. Konfigurace udává polohu objektu *veh* v rámci membránové struktury. Vzhledem k tomu, že systém obsahuje pouze jediný objekt, je možné na základě po sobě jdoucích konfigurací systému vysledovat, která pravidla byla použita a kterými synapsemi byl objekt přemísťován. Pohyb mezi membránami  $1 \rightarrow 2$ ,  $2 \rightarrow 3$  a  $3 \rightarrow 1$  označíme jako pohyb kladným směrem (+) a pohyb  $1 \rightarrow 3$ ,  $3 \rightarrow 2$  a  $2 \rightarrow 1$  jako pohyb záporným směrem (-). Počáteční pozice objektu *veh* je membrána 1.

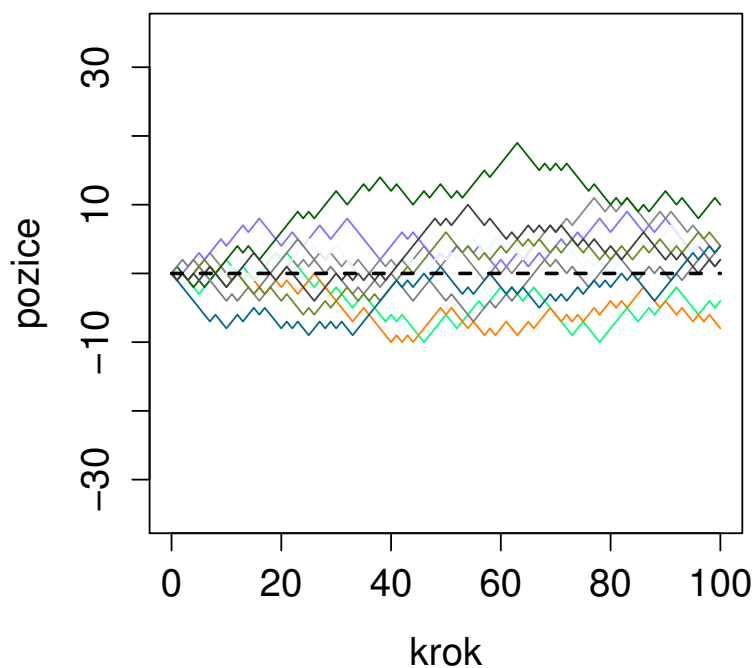
Krok	Membrána 1	Membrána 2	Membrána 3	Směr pohybu
1	0	0	1	$\rightarrow -1$
2	1	0	0	$\rightarrow +1$
3	0	0	1	$\rightarrow -1$
4	0	1	0	$\rightarrow -1$
5	1	0	0	$\rightarrow -1$
6	0	0	1	$\rightarrow -1$
7	1	0	0	$\rightarrow +1$
8	...	...	...	...

Tabulka 6: Výsledek simulace P systému pro náhodnou vycházku. Přítomnost objektu v membráně je označena jako binární proměnná 0/1

Na základě pozice objektu *veh* v po sobě jdoucích krocích je dopočítána cesta

objektu systémem. Předpokládáme, že na počátku výpočtu je pozice objektu 0 a každým krokem se posune o jednotku v kladném nebo záporném směru. Výsledky 20 běhů programu po dobu 100 kroků jsou na obrázku 17.

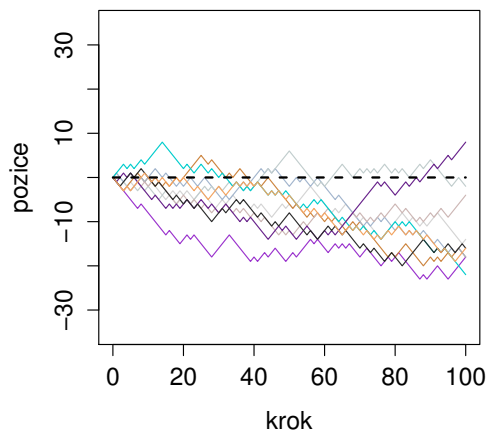
### 10 náhodných procházek, $p=0.50$



Obrázek 17: 10 náhodných vycházek

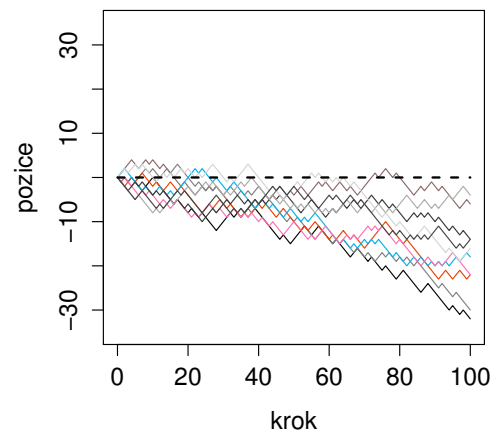
Změnou pravděpodobností  $p$  a  $q$  lze modelovat různé scénáře, jak ukazují obrázky 18 a 19. Simulace, kdy nejsou pravděpodobnosti cest  $p$  a  $q$  shodné však již nejsou náhodnými procházkami a zasahují spíše do oblasti teorie her.

**10 náhodných procházek,  $p=0.45$**



Obrázek 18: 10 náhodných procházek s pravděpodobnostmi  $p = 0,45, q = 0,55$

**10 náhodných procházek,  $p=0.40$**



Obrázek 19: 10 náhodných procházek s pravděpodobnostmi  $p = 0,40, q = 0,60$

Model navržený pro simulace náhodné procházky obsahuje pouze tři membrány, lze jej však rozšířit tak, aby umožnil simulace náhodných procházek v libovolném grafu. Lze například sledovat, kolik náhodných kroků lze v grafu učinit bez opakovaného použití synapse. V rámci navrženého modelu *transportních P systémů* lze použít pro simulaci takového chování následující pravidlo:

$$\left[ veh \mid syn_{(i,j)}^+ \right]_i \xrightarrow[s^+ \rightarrow s^-]{t=1, p} \left[ veh \right]_j. \quad (43)$$

Během aplikace pravidla dojde k přesunutí objektu *veh* z membrány *i* do membrány *j*, pokud existuje otevřená synapse  $syn_{(i,j)}^+$ . Při aplikaci pravidla dojde ke změně stavu synapse  $s^+ \rightarrow s^-$ . Laicky řečeno tedy objekt *veh* při cestě synapsí  $(i, j)$  tuto synapsi uzavírá. Pokud je do náhodné membrány umístěn objekt *veh*, výpočet probíhá do té doby, než se objekt *veh* dostane do membrány, z níž nevedou žádné otevřené synapse. Počet kroků výpočtu odpovídá počtu navštívených synapsí (za předpokladu jednotkové délky *t* transportu synapsí).

Zde navržený model předpokládá, že prostor, v němž je náhodná procházka realizována, je neohraničený. Pokud je prostor ohraničený, jedná se o náhodnou procházku s pohlcujícími nebo odrážejícími stěnami. Příkladem takové náhodné procházky je ruinování hráče, kerému se věnuje podkapitola 4.8.

Další problémy, často zkoumané v teorii grafů, lze v *transportních P systémech* simulovat při rozšíření modelu o další prvky. Například lze zkoumat *first-hitting time*, což je počet kroků, než náhodně se pohybující agent v síti navštíví uzel, v němž již byl. P systémy poskytují mnoho způsobů, jak podobný model navrhnout, nicméně všechny vyžadují rozšíření nad rámec *transportních P systémů*. Jedním ze způsobů je například zavedení polarizace membrán, kdy již navštívená membrána bude označena jinou polarizací než membrána nenavštívená. Jiný způsob je umístit do každé membrány jeden objekt, reprezentující zdroje/energii. Při opuštění membrány objektem *veh* by byly zdroje spotřebovány a pokud by se dostal objekt *veh* do membrány, kterou již navštívil, výpočet by skončil, protože by v membráně nebyly přítomné zdroje, které by objektu *veh* umožnily ji opustit.

Lze navrhnout řadu variant P systémů, které budou řešit dílčí úlohy, spojené s teorií grafů. V případech jako jsou výše zmíněné *first-hitting time* nebo sledování projití všech synapsí však P systémy nevyužívají paralelismus výpočtu a spíše simulují chování agenta, pohybujícího se sítí. Největší potenciál využití P systémů v oblasti teorie grafů je při řešení problémů, v nichž dochází k současnému pohybu většího

počtu agentů sítí. Otázky typu: „*Jak dlouho trvá x náhodně se pohybujícím objektům, než navštíví všechny uzly/hrany grafu?*“ jsou ideálním zadáním pro P systémy, které umožňují paralelní modelování jevů této povahy.

## 4.8 Simulace: ruinování hráče

Ruinování hráče (*Gambler's ruin*) je statistický koncept, který je nejčastěji vyjádřen jako pravděpodobnost, že hráč s počátečním jměním a konstatním vkladem skončí s nulou za předpokladu, že hráč ukončí hru jen pokud má celý bank, nebo nemá nic.

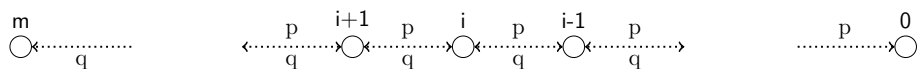
Předpokládejme, že hrají dva hráči ( $a$  a  $b$ ), jejichž počáteční jmění je  $A$  a  $B$  jednotek, vklad je jedna jednotka a pravděpodobnosti, že hráči vyhrají jedno kolo hry jsou  $p$  a  $q = 1 - p$ .

Pravděpodobnost, že hráč  $a$  bude zruinován lze vypočíst jako (Epstein, 2010):

$$p(a) = \begin{cases} \frac{1 - (\frac{q}{p})^B}{1 - (\frac{q}{p})^{A+B}} & \text{pokud } p \neq q \\ \frac{B}{A+B} & \text{pokud } p = q \end{cases} \quad (44)$$

Při každém kole hry se hraje o jednu jednotku jmění a to do té doby, než je jeden z hráčů zruinován (a druhý vlastní vše). Za předpokladu, že pravděpodobnosti jsou shodné ( $p = q$ ) je modelem tohoto problému jednoduchá náhodná procházka. V rámci *transportních P systémů* lze ruinování hráče modelovat následovně:

Označme jako  $N = A + B$  celkové jmění obou hráčů. Počet membrán membránového systému je  $m = N + 1$  a popisky membrán jsou  $0, \dots, m$ . Membrány  $i \in \{1, \dots, m-1\}$  jsou synapsi spojeny vždy se dvěma membránami  $i-1, i+1$ . Délky všech synapsí jsou rovny jedné. Pravděpodobnosti asociované se synapsi jsou  $p$  pro synapsi  $(i, i+1)$ , resp.  $q = 1 - p$  pro synapsi  $(i, i-1)$ . Z membrán  $0$  a  $m$  nevedou žádné synapse, nicméně existují synapse  $(1, 0)$  a  $(m-1, m)$ .



Obrázek 20: Membránová struktura pro simulaci ruinování hráče

Na počátku výpočtu neobsahuje žádná z membrán objekty s výjimkou té membrány, jejíž popisek  $i$  odpovídá hodnotě  $A$  jmění jednoho z hráčů. Tato membrána obsahuje jeden objekt *veh*.

Membrány  $i \in \{1, \dots, m - 1\}$  obsahují pravidla ve tvaru:

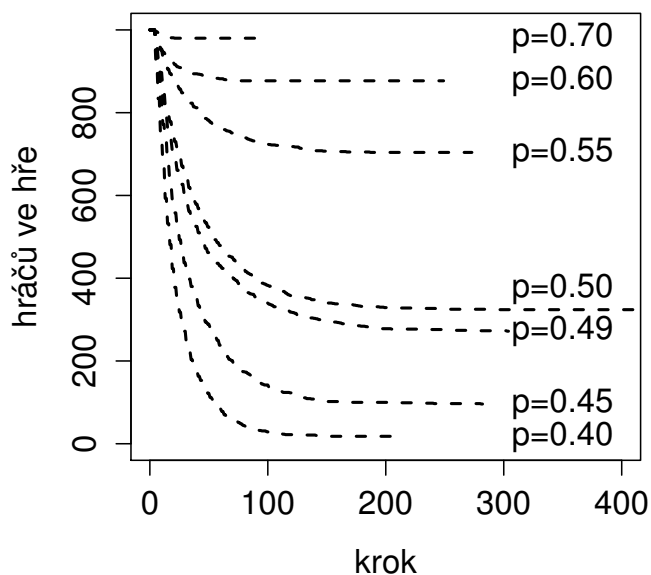
$$\left[ veh \mid syn_{(i,j)} \right]_i \xrightarrow{t=1,p} \left[ veh \right]_j, \quad (45)$$

což je stejné pravidlo, které obsahuje systém pro simulaci náhodné procházky. Membrány  $0, m$  neobsahují žádná pravidla.

Výpočet systému pokračuje do té doby, než je objekt  $veh$  přemístěn do některé z membrán  $0, m$ . Pak je výpočet ukončen, protože nejsou žádná pravidla, která by mohla být v systému aplikována.

Výhodou použití P systému pro simulaci ruinování hráče je paralelismus výpočtního modelu. Do zdrojové membrány může být umístěn libovolný počet  $n$  objektů  $veh$  a výsledek výpočtu bude odpovídat  $n$  simulacím systému s jediným hráčem. Vzhledem k tomu, že nezáleží na průběhu konkrétní hry (posloupnosti membrán, které jsou objektem navštíveny), ale pouze výsledku (hráč vše vyhrál nebo vše prohrál), lze P systémy pro tento účel dobře použít.

### ruinování hráče, 1000 běhů simulac



Obrázek 21: Simulace ruinování hráče za použití různých pravděpodobností

Výsledky simulací 1000 her s rozdílnými nastaveními pravděpodobnosti výhry jsou zobrazené na obrázku 21. Počáteční jmění hráče  $a$  je pět jednotek, hráče  $b$  deset jednotek. Na ose  $x$  je vynesena počet simulací, v nichž hráč  $a$  je stále ve hře, na ose  $y$  je počet odehraných kol hry. Maximální hodnota na ose  $x$  odpovídá maximální



délce hry, zaznamenané během tisíce simulací. Je patrné, že čím menší rozdíl je mezi pravděpodobnostmi  $p$  a  $q$ , tím je hra delší.

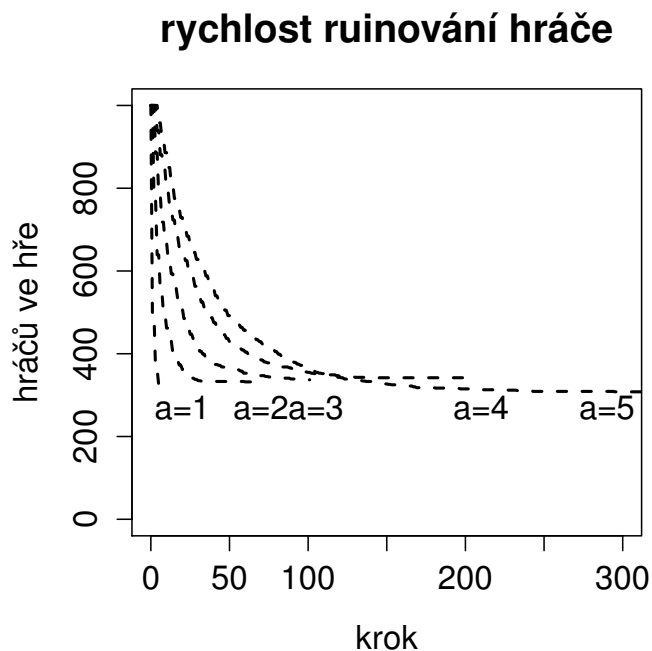
Podíl počtu her, ve kterých vyhrál hráč  $a$  lze porovnat s očekávanými hodnotami, vypočtenými dle vzorce 44 (počáteční jmění hráče  $a$  je 5, hráče  $b$  je 10).

Pravděpodobnost	0,10	0,20	0,30	0,40	0,45	0,49	0,50	0,55	0,60	0,70
Očekáváno (%)	0,0	0,0	0,0	1,5	9,0	26,9	33,3	66,7	87,0	98,6
Pozorováno (%)	0,0	0,0	0,0	1,7	9,6	27,2	32,3	70,4	87,7	98,0

Tabulka 7: Výsledek simulace ruinování hráče, 1000 her, počáteční jmění 5 a 10

Hodnoty nasimulované pomocí P systémů odpovídají hodnotám vypočteným. Kromě pravděpodobnosti zruinování, která lze exaktně vypočíst a není nutné ji simulovat, můžou simulace sloužit ke zkoumání například délky trvání hry. Pokud se jedná o férovou hru ( $p = q = 0,5$ ) a tedy vlastně náhodnou procházku, zůstává pravděpodobnost zruinování hráče stejná, pokud se nemění poměr počátečního jmění. Pravděpodobnost zruinování je stejná, pokud je počáteční jmění hráčů 1 a 2, 2 a 4, 5 a 10 apod.

Co se mění, je rychlost ruinování, jak ukazuje obrázek 22.



Obrázek 22: Rychlost ruinování hráče. Počáteční jmění prvního hráče je označeno jako  $a$ , jmění druhého hráče je dvojnásobné

Jak uvádí (Steele, 2001), očekávané trvání hry lze vypočíst jako

$$p(a) = \left\{ \frac{A}{q-p} - \frac{A+B}{q-p} \times \frac{1-\frac{q}{p}A}{1-\frac{q}{p}A+B} A \times B \right. \quad \text{pokud } p = q \quad (46)$$

To je pozoruhodné, poněvadž při férové hře, kdy oba hráči mají stejnou pravděpodobnost výhry a počáteční jmění hráčů je  $A = 1$ ,  $B = 99$  je očekávané trvání hry 99 kol ( $A \times B = 99$ ). Přitom šance, že hráč bude zruinován již po jednom kole, je 50 %.

Porovnání očekávané délky trvání hry a rozdělení délky her u 1000 simulací ukazuje následující tabulka

A	B	Očekávané trvání	Kvantil simulovaného rozdělení						
			50	75	90	95	97,5	99	100
1	2	2	2	2	4	5	6	7	14
2	4	8	6	10	16	22	28	32	42
3	6	18	13	24	38	49	59	75	145
4	8	32	22	42	68	88	100	122	286
5	10	50	38	72	115	139	168	195	279
1	99	99	3	9	59	311	1092	2773	6751

Tabulka 8: Délka hry, 1000 her,  $p = q$

Prvních 5 řádků tabulky obsahuje výsledky simulací, kdy poměr  $\frac{A}{B} = \frac{1}{2}$ . Mění se pouze absolutní hodnota parametrů  $A$  a  $B$ . Z výsledků je zřejmé, že charakteristiky rozdělení délky trvání rostou rychleji než hodnoty  $A$  a  $B$ .

V posledním řádku tabulky jsou výsledky simulace při nastavení  $A = 1$ ,  $B = 99$ , přičemž očekávaná hodnota délky hry je 99 kol. To odpovídá 92% kvantilu rozdělení, získaného simulací s 1000 hráči.

Ruinování hráče představuje jednoduchý a dobře popsáný problém, kde řada otázek typu „*Jaká je nejkratší možná doba hry?*“, „*Jaká je pravděpodobnost, že hra skončí do pěti kol?*“, „*O kolik větší jmění musí hráč mít, aby kompenzoval o 10 % nižší šanci výhry?*“ může být zodpovězena analyticky a simulace nejsou potřeba. V případě složitějších zadání, kdy se pravděpodobnost výhry mění na základě aktuálního jmění, mění se výška sázky apod. mohou být simulace užitečné a díky paralelnosti výpočtu představují P systémy zajímavou alternativu k jiným výpočetním metodám, založeným na opakované simulaci hry jednoho hráče.

Reálné systémy, v nichž jsou principy náhodné procházky (ať již férové či neférové) zřídka odpovídají jednoduchému modelu, kdy jsou jak pravděpodobnosti  $p, q$ , tak i

velikost sázky konstantní po celou dobu výpočtu. Návrh P systémů je variabilní a umožňuje vytvořit model systému, který tyto prvky obsahuje.

## 4.9 Transportní P systém s vozidly

### 4.9.1 Formální definice

Transportní P systém s vozidly je konstrukt

$$\Pi = (O, C, \mu, w_{m1}, \dots, w_{mk}, w_{v1}, \dots, w_{vl}, syn, R, R_p, i_o), \quad (47)$$

kde

- $O$  je abeceda objektů,
- $C \subset O$  je abeceda katalyzátorů,
- $\mu = \{m1, \dots, mk, v1, \dots, vl\}$  je množina membrán s  $k$  stanicemi a  $l$  vozidly. Stanice se nachází přímo ve vnějším prostředí, membrány reprezentující vozidla jsou elementární a nachází se ve stanicích,
- $w_{m1}, \dots, w_{mk}, w_{v1}, \dots, w_{vl}$  jsou multimnožiny objektů, nacházejících se na počátku výpočtu v jednotlivých stanicích  $m1, \dots, mk$  a vozidlech  $v1, \dots, vl$ ,
- $syn \subseteq \{(i, j, t) | i, j \in \{m1, \dots, mk\}, i \neq j, t \in \mathbb{N}^+\}$  je množina synapsí mezi membránami. Každá synapse obsahuje informaci o času  $t$ ,
- $R_{m1}, \dots, R_{mk}, R_{v1}, \dots, R_{vl}$  je množina pravidel asociovaných s jednotlivými membránami systému,
- $R_p$  je množina visících (*pending*) pravidel,
- $i_o$  je buď jeden z regionů  $m1, \dots, mk, v1, \dots, vl$  a pak je tento region výstupním regionem systému, nebo je 0 a pak je výsledek výpočtu uložen ve vnějším prostředí systému.

Pravidla mají následující tvar:

- $(xk, in; y, out)$  je antiportní pravidlo, kde  $x, y \in O$  jsou objekty (nikoli multimnožiny objektů) a  $k \in C$  je katalyzátor. Pravidlo popisuje situaci, kdy objekt  $x$  je přemístěn dovnitř membrány, zatímco objekt  $y$  ji ve stejném kroku opustí. Objekt  $k$  slouží katalyzátor a je použit standardním způsobem.
- $(xk, ind)$  je uniportní pravidlo, kde  $x \in O$  je objekt a  $ind \in \{in, out, here\}$  je indikátor cíle. Pokud  $ind = in$ , objekt je přesunut do membrány vnitřní, pokud

$ind = out$ , objekt je přesunut do membrány rodičovské (pokud je současná membrána vnější membránou, je objekt přesunut do vnějšího prostředí) a pokud  $ind = here$ , pak objekt zůstává v současné membráně. Objekt  $k \in C$  slouží jako katalyzátor a je použit standardním způsobem.

- $([ ], in_j, t)$  je pravidlo, popisující přemístění vnitřní membrány (označené jako  $[ ]$ ) do membrány  $j$ . Podmínkou je existence synapse mezi membránou, v níž je pravidlo umístěno a membránou  $j$ . Přemístění membrány zabere  $t$  časových jednotek.
- $([ ], \delta)$  je pravidlo, popisující rozpuštění vnitřní membrány.
- $([ ], \epsilon)$  je pravidlo, popisující vytvoření membrány uvnitř současné membrány.

#### 4.9.2 Popis modelu

Pravidla představeného modelu navazují na komunikační P systémy, P systémy s aktivními membránami, P systémy s mobilními membránami a P systémy s nosiči. Od všech těchto modelů se však model *transportních P systémů s nosiči* v některých aspektech odlišuje.

Zde použitý zápis využívá symbolů  $\delta$  a  $\epsilon$  pro označení operací rozpuštění, respektive vytvoření membrány.

Zatímco v modelu *transportních P systémů* byly objekty transportovány přímo mezi membránami, které se nacházely ve vnějším prostředí, zde jsou k transportu použity membrány, které samy mění polohu a přemísťují se. Existují tedy dva druhy membrán – membrány umístěné v uzlech sítě (*stanice*), kterými se do a ze systému dostávají objekty a membrány mobilní, které jsou mezi těmito membránami transportovány. Mobilní membrány slouží jako *vozidla (nosiče)*, která umožňují transport objektů. *Vozidla* mohou být ve *stanicích* vytvářena a mohou v nich zanikat, zatímco *stanice* jsou pevně dané a po celou dobu výpočtu neměnné.

Katalyzátory jsou nutné s ohledem na práci s kapacitou vozidel. Katalyzátory se nacházejí v mobilních membránách a jejich počet odpovídá maximálnímu počtu objektů, které mohou být *vozidlem* transportovány.

Vzhledem k tomu, že mobilní membrány vznikají a jsou rozpouštěny, je nutné, aby katalyzátory byly v membránách přítomny v okamžiku vytvoření membrány. V případě rozpuštění membrány jsou všechny objekty v ní přítomné přesunuty do membrány vnější; pokud taková membrána neexistuje, tak do vnějšího prostředí.

Synapse, stejně jako v *transportních P systémech*, obsahují informaci o čase, který trvá vykonání cesty skrz synapsi. Obdobně jako v modelu představeném v kapitole 4.1 jsou použita visící pravidla, přičemž algoritmus aplikace visících pravidel zůstává stejný. Jediná pravidla, která obsahují informaci o čase, jsou pravidla popisující přemístění mobilních membrán. Čas vykonání ostatních pravidel je jednotkový. Z pravidla  $([ ], in_j, t)$  vzniká visící pravidlo  $([ ], in_j, t - 1)$ . Čas visícího pravidla je v každém kroku snížen o jednu jednotku a v případě, že je na začátku výpočetního kroku nulový, je v membráně  $j$  vytvořena membrána  $[ ]$  včetně celého svého obsahu - objektů i pravidel.

Navržený model obsahuje membrány ve dvou úrovních. Ve vnějším prostředí jsou obsaženy membrány reprezentující *stanice* a v nich jsou obsaženy membrány reprezentující *vozidla*. Vnější prostředí nikdy nekomunikuje přímo s *vozidly*. Bylo by tedy možné systém navrhnout tak, že všechna pravidla by byla obsažena v membránách – *stanicích* – a tato pravidla by popisovala jak transport objektů z vnějšího prostředí do *stanic*, tak ze *stanic* do *vozidel* a naopak. Takový systém by byl dostatečný za předpokladu, že přepravované objekty cestují vždy pouze mezi dvěma sousedními membránami. To však nemusí být pravda – objekt může vstoupit do mobilní membrány a setrvat v ní libovolně dlouhou dobu. Musí tedy existovat pravidlo, které popisuje setrvání objektu v mobilní membráně. Tvar tohoto pravidla je  $(xk, here)$  a popisuje setrvání objektu  $x$  v současné membráně za předpokladu, že je v ní přítomen katalyzátor  $k$ .

Stejně jako v případě P systémů se symportem/antiportem jsou přítomná pravidla komunikační. Objekty jsou v rámci systému přemístovány, ale nemění se jejich množství. Může se však změnit jejich typ (objekt  $u$  se během transportu může změnit v objekt  $v$ ) Jediný způsob, jak změnit množství objektu  $u$  v systému, je přivést jeho další kopie z vnějšího prostředí. Pravidla typu  $u \rightarrow uu$  a  $uu \rightarrow u$ , která by popisovala změnu množství objektů v systému, nejsou přítomna.

Mobilní membrány, pokud jsou vytvářeny, musí být vytvářeny včetně objektů (katalyzátorů) a pravidel, což zde představený model odlišuje od P systému s mobilními membránami (Krishna a Păun, 2005; Aman a Ciobanu, 2008).

Model *transportních P systémů s nosiči* se nejvíce blíží *P systémům s nosiči* (Martín-Vide et al., 2002b), proto i podobnost názvů. P systémy s nosiči používají podobný princip pohybu objektů – vytvoření membrány, v níž jsou objekty přesunuty a její následné rozpuštění. Nosiče však slouží pouze k transportu „vertikálně“ – do membrán vnitřních nebo vnějších. Zde představený model přemísťuje nosiče „horizontálně“ – mezi membránami na stejné úrovni, které jsou spojeny synapsemi.

Druhý podstatný rozdíl spočívá v tom, že nosič není po vstupu do další membrány rozpuštěn, ale dále existuje. V *P systémech s nosiči* (Martín-Vide et al., 2002b) vždy nosič po transportu mezi dvojicí membrán zaniká, ve zde představených *transportních P systémech s nosiči* dále existuje a může se membránovou strukturou pohybovat po celou dobu výpočtu. Z tohoto rozšíření plyne další významná vlastnost – objekty mohou v nosičích zůstat po libovolně dlouhou dobu. Objekt může být zapouzdřen do nosiče v membráně  $i$ , opustit nosič v membráně  $k$  a mezitím být přemístěn libovolným počtem dalších membrán.

### 4.9.3 Použití a varianty

Největší výzvou při modelování dopravy pomocí P systémů je reprezentace voleb, které účastníci během přepravy činí. V hromadné dopravě, realizované pomocí velkokapacitních přepravních prostředků, uvažujeme jen volby cestujících, nikoli volby vozidel. Předpokládáme tedy, že trasy jízdní řád a kapacity vozidel jsou dány na začátku výpočtu.

Jsou tři volby, které cestující v průběhu přepravy činí:

- *Nastoupím do vozidla, nebo setrvám ve stanici* – tuto volbu činí cestující, který je ve stanici, do níž přijede vozidlo
- *Setrvám ve vozidle, nebo z něj vystoupím* – tuto volbu činí cestující ve vozidle, které přijíždí do stanice
- *Setrvám ve stanici, nebo ji opustím* – tuto volbu činí cestující, který vystoupil z vozidla. Toto rozhodnutí je relevantní jen u systémů, které obsahují přestupní stanice. Pokud je přepravní systém tvořen linkami, které se nekříží (např. okružní linky), pak nemá cenu se tímto typem rozhodnutí zabývat.

Všechny tři výše vypsané rozhodovací procesy mohou být řešeny pomocí pravděpodobností. Například situace „*Setrvám ve vozidle, nebo z něj vystoupím*“ je rozhodnutí mezi dvěma pravidly:

- $(x, out)$  – vystoupení z vozidla
- $(x, here)$  – setrvání ve vozidle

Každé z těchto pravidel může mít přiřazenou pravděpodobnost  $p$ . Při modelování jednoduchých systémů nebo simulaci teoretických modelů může být tento způsob

dostačující, při modelování reálných systémů však bude zřejmě vznikat problém v realistickém stanovení pravděpodobností; odvození tras cestujících na základě OD matice není triviální úkol (Sherali a Park, 2001).

Alternativní způsob řešení rozhodovacích procesů spočívá v přiřazení cesty každému objektu, který se transportu účastní. Namísto objektu  $x$  bude existovat objekt  $x_{k,\dots,r}$ , kde  $k, \dots, r$  jsou označení stanic, které musí objekt během své cesty navštívit. Stanice  $r$  je stanicí konečnou.

Seznam stanic reprezentující cestu vozidla musí být přiřazen i membránám, reprezentující vozidla.

Pravidla, popisující rozhodnutí o setrvání ve vozidle se změní na:

- $(\{\{x_{k,l,\dots,r}\}_j\}_i, out)$  – vystoupení z vozidla
- $(\{\{x_{j,k,l,\dots,r}\}_j\}_i, here)$  – setrvání ve vozidle

Jako  $\square_i$  je zde označena membrána, v níž se vozidlo nachází. Vozidlo je označeno jako  $\{\}_j$ , kde  $j$  je popisek následující stanice. Objekt  $x_{k,l,\dots,r}$  čteme jako objekt, jehož cílovou stanicí je  $r$  a jehož trasa je posloupností stanic  $k, l, \dots, r$ . Pokud se příští stanice vozidla  $j$  shoduje s první příští stanicí objektu ( $j = k$ ), pak objekt (cestující) setrvává ve vozidle. V opačném případě systém opouští.

Objekt cestujícího začíná jako  $x_{k,l,m,\dots,r}$  a v každé projeté stanici je jeho trasa aktualizována. Ve stanici  $k$  se z něj stane objekt  $x_{l,m,\dots,r}$ , ve stanici  $l$  se z něj stane  $x_{m,\dots,r}$  atd. Jakmile neexistuje příští stanice, tzn. má tvar  $x_{\emptyset}$ , opouští objekt vozidlo (a posléze systém).

Zjednodušeně by bylo možné cestujícího reprezentovat jako objekt  $x_{k,r}$ , kde  $k$  je příští stanice a  $r$  je konečná stanice. Příští stanici lze v každé stanici na základě aktuální pozice a cíle přepočítat.

Stanice tvoří orientovaný graf a hledání cesty v grafu je triviálním úkolem. Matici, která pro každou dvojici stanic *aktuální – cílová* stanovuje následující stanici lze navíc vypočítat na počátku výpočtu. Vzhledem ke statické síti stanic se matice nebude měnit. Při takto navrženém modelu však není možné pracovat s různými cestami mezi dvěma body, neboť pro každou dvojici stanic *aktuální – cílová* existuje přesně jedna příští stanice (přepokládáme zde, že mezi každou dvojicí stanic v síti existuje cesta; případ, kdy jsou některé stanice nedostupné z jiných stanic, zde neuvažujeme). Tento způsob výpočtu navíc vyžaduje, aby mimo rámec modelu P systémů probíhal výpočet příštích stanic. Pokud je použit zápis pomocí celé plánované trasy, je možné všechny výpočty reprezentovat pomocí pravidel zde navrženého modelu.



Z hlediska realistického zachycení transportu je nutné zabývat se posloupností operací při nástupu a výstupu ve stanici. Dochází zde ke třem operacím: cestující vystupují z vozidla, cestující zůstávají ve vozidle a cestující nastupují do vozidla. Tyto operace jsou reprezentovány jako pravidla:

- $(\{x_{k,l,\dots,r}\}_j)_i, out)$  – vystoupení z vozidla, pravidlo je přítomno ve vozidle
- $(\{x_{j,k,l,\dots,r}\}_j)_i, here)$  – setrvání ve vozidle, pravidlo je přítomno ve vozidle
- $(\{x_{j,k,l,\dots,r}\}_j, in)$  – nástup do vozidla, pravidlo je přítomno ve stanici

Tento design pravidel je nepraktický ze dvou důvodů. Jednak jsou operace, které na sebe logicky navazují, řízeny množinou pravidel ve dvou různých membránách. Z hlediska modelu jako takového to není překážkou, ale softwarová implementace modelu, kde jeden proces je řízen ze dvou míst, je nešikovná. Především však tento návrh neřeší problém, že nejprve by měl proběhnout výstup a pak nástup a nepočítá se zde s kapacitou vozidla.

Upravený tvar pravidel, která tohle řeší, je:

- $(\{x_{k,l,\dots,r}\}_j)_i, out)$  – vystoupení z vozidla, pravidlo je přítomno ve vozidle, není vyžadován katalyzátor.
- $(\{x_{j,k,l,\dots,r}, k'\}_j)_i, here)$  – setrvání ve vozidle, pravidlo je přítomno ve vozidle, je vyžadována přítomnost katalyzátoru ( $k'$ )
- $(x_{j,k,l,\dots,r}\{k'\}_j)_i, in)$  – nástup do vozidla, pravidlo je přítomno ve vozidle, je vyžadována přítomnost katalyzátoru ( $k'$ )

Tato pravidla již pracují správně s kapacitami vozidla a jsou všechna umístěna ve vozidle. Celý proces je tedy řízen jednou sadou pravidel, přítomných v jedné membráně. Pravidla pracující s katalyzátory jsou ta, která popisují setrvání ve vozidle a nástup cestujících. V reálné dopravě má setrvání ve vozidle preferenci před nástupem (nastupující lidé zaplňují ta prázdná místa, která jsou k dispozici). Paralelní výpočet P systémů v tomto představuje problém, protože z obou pravidel – *setrvání ve vozidle* a *nástup* je vybíráno náhodně. Je proto nutné zavést prioritu pravidel. Pravidlo popisující *setrvání ve vozidle* má vyšší prioritu než *nástup*, je vykonáno tolikrát, kolikrát je možné a až pak je vykonáno pravidlo popisující *nástup*. Pravidlo popisující *výstup* nemusí mít přiřazenou prioritu, protože nedochází ke konfliktu s žádným

z dalších pravidel. Všichni cestující již přítomní ve vozidle buď setrvávají nebo vystupují, přičemž operace je zvolena na základě následující stanice a operace *výstup* nespotežbovává katalyzátory.

Rozhodnutí, zda cestující nastoupí do vozidla, je popsáno pravidlem pro nástup do vozidla. Toto pravidlo je aplikovatelné pouze pokud příští stanice vozidla  $\{\}_j$  odpovídá příští stanici cestujícího  $x_{j,k,l,\dots,r}$ . Pokud se liší, pravidlo nemůže být vykonáno a cestující setrvává ve stanici, i když je v ní přítomné vozidlo.

Transport vozidel popisuje pravidlo ve tvaru:

- $(\{\}_j]_i, in_j, t, k)$  – vozidlo, jehož následující stanice je  $j$  a které se nachází ve stanici  $i$ , je přesunuto do stanice  $j$ . Délka vykonání pravidla je  $t$  časových jednotek. Symbol  $k$  značí stanici, do jíž bude vozidlo přemístěno z membrány  $j$ .

Tento zápis je komplikovaný, nicméně nutný k zachycení jízdního řádu vozidel. Nejjednodušší zápis pravidla pro přesun membrán je  $(\{\}_j]_i, in_j, t)$ ; tento zápis však umožňuje vozidlu cestovat libovolným směrem z membrány  $j$ . Jakmile je vozidlo  $\{\}$  ve stanici  $j$ , nelze poznat, z jaké stanice přišlo a tudíž nelze zachytit jízdní řád – nelze poznat, do které stanice má být přesunuto dále. Vozidlo by se tak mohlo sítí pohybovat zcela náhodně, což neodpovídá reálné dopravě.

Po aplikování pravidla  $(\{\}_j]_i, in_j, t, k)$  a uplynutí  $t$  časových jednotek dojde k vytvoření vozidla  $\{\}_k$  v membráně  $j$ . Obsah vozidla (objekty a pravidla) je beze změny zkopírován.

Vytvoření a zánik membrán probíhá podle pravidel ve tvaru:

- $(\{\}_\emptyset]_i, \delta)$  – vozidlo, které nemá určenou následující stanici a nachází se v membráně  $i$ , je rozpuštěno. Objekty v něm přítomné se přesunou do membrány  $i$  (jedná se o rozpuštění tak, jak funguje v běžných P systémech)
- $(\{\}_j]_i, \epsilon)$  – v membráně  $i$  je vytvořeno vozidlo, jehož následující stanicí je  $j$ .

Vozidla v okamžik svého vytvoření musí obsahovat všechna pravidla, která popisují nástup, výstup a setrvání objektů a rovněž počet  $n$  objektů  $k'$ , které reprezentují kapacitu vozidla.

Posledním procesem, který model musí vystihnout, je příchod a odchod cestujících do stanic. Je reprezentován pomocí pravidel:

- $([x_{j,k,l,\dots,r}, k']_i, in)$  – příchod cestujících do stanice. Ve stanici  $i$  je vytvořen objekt  $x_{j,k,l,\dots,r}$ , reprezentující cestujícího, jeho příští stanicí je  $j$  a který cestuje do stanice  $r$ . Pro aplikaci pravidla je nutné, aby byl ve stanici přítomen objekt  $k'$ , reprezentující katalyzátor.
- $([x_{\emptyset}]_i, out)$  – cestující, který nemá určenou příští stanicí a nachází se ve stanici  $i$ , je odstraněn ze systému (odeslán do vnějšího prostředí).

Každá stanice obsahuje určený počet katalyzátorů  $k'$ , který reprezentuje počet cestujících, kteří stanicí vstoupí do systému v jednom časovém kroku. Reálná dopravní poptávka není v čase konstantní, její cykly odpovídají lidské aktivitě (Kitamura et al., 2000) a pro větší realističnost modelu je žádoucí, aby se počet cestujících, přicházejících do stanice, v čase měnil. Jednoduchý způsob odhadu dopravní poptávky je vypočítat jádrovou hustotu frekvence jízdy vlaků na základě jízdního řádu. Za předpokladu, že jízdní řády jsou konstruovány tak, aby uspokojily dopravní poptávku, by mělo platit, že frekvence vlaků na dané lince odpovídá poptávce ze straně cestujících (Janoška a Dvorský, 2013). Ze znalosti jízdního řádu pak lze zpětně odvodit rozložení dopravní poptávky v čase.

#### 4.9.4 Limity

Zde představený model transportu pomocí *transportních P systémů s nosiči* při snaze o co nejrealističtější zachycení studovaného jevu ztrácí vlastnosti, které jsou vlastní P systémům, zejména nedeterminismus. Výše popsaný model neobsahuje nedeterminismus v žádné své části, jeho jediným zdrojem mohou být cílové stanice, přiřazené jednotlivým objektům. Při zachování nedeterminismu v rozsahu, který je uveden v základní variantě modelu (podkapitola 4.9.1), není navržený systém schopen realisticky reprezentovat dopravu v systémech hromadné dopravy. Jízdní řády a předdefinované cesty cestujících nelze v rámci P systémů věrohodně reprezentovat a zachovat přítomný nedeterminismus systému.

*Transportní P systém s nosiči* proto není vhodným nástrojem pro simulaci transportu cestujících v systémech hromadné dopravy.

## 5 VÝSLEDKY

Tato práce představuje dva modely transportu v P systémech. Oba simulují transport objektů v předem dané síti, která je statická. Modely nejsou schopny zachytit transport popsáný například vektory pohybu (směr, vzdálenost), dokáží pouze přesouvat objekty v síti, jejíž uzly mohou být lokalizován v prostoru. Pro simulaci jevů, dějících se ve spojitém prostoru je možné uspořádat síť membrán do pravidelné mřížky s konstantní délkou synapse mezi sousedícími membránami. Tento princip práce s prostorem byl použit v práci (Barbuti et al., 2011) jako takzvané *prostorové P systémy*. Zde navržené modely aspirují modelovat jiný druh transportu.

Představené modely předpokládají, že jediné lokalizované prvky systému jsou membrány. Pro objekty platí pravidlo, že jsou v rámci membrán neuspořádané a jsou vzájemně zaměnitelné. Membrány reprezentují v prostoru lokalizované entity reálného světa, jejich absolutní poloha v rámci daného souřadného systému se výpočtu neúčastní, může však být atributem membrán. Z hlediska reprezentace prostoru v geografických informačních systémech odpovídají navržené varianty P systému vektorovému modelu dat, zatímco *prostorové P systémy* odpovídají rastrovému modelu dat.

Oba zde navržené modely jsou postaveny na podobném schématu, kdy membrány jsou rozmístěny v grafu a spojeny synapsami, podobně jako v *tkáňovitých P systémech* (Martín-Vide et al., 2002a). Existence synapse vyjadřuje sousedství membrán (možnost přímo cestovat mezi danou dvojicí). Ačkoli při návrhu modelu bylo uvažováno, že synapse představují komunikace spojující místa v síti, nemusí platit, že synapse reprezentují entity reálného světa. Synapse nejsou fyzicky existujícími komunikačními cestami (silnicí, vodním tokem, potrubím apod.), ale pouze mentálním konstruktem, který stanovuje, že mezi dvěma membránami lze přímo cestovat.

Vzdálenost mezi membránami může být vyjádřena pomocí pravděpodobnosti nebo času. V případě modelování například disperze populace může pravděpodobnost synapse být inverzní hodnotou vzdáleností, tudíž vzdálenější membrány budou mít nižší pravděpodobnosti výběru. Vhodnější způsob vyjádření vzdálenosti je však pomocí času, který transport mezi dvojicí membrán zabere. Například pro simulaci transportu vozidel v silniční síti je tato reprezentace dostačující, pro simulaci dálkové migrace živočichů zřejmě nikoli. Způsob, jak v rámci P systémů zahrnout vzdálenost (mezi membránami či objekty) přímo do výpočtu nebyla zatím v literatuře publikována. Z pohledu P systémů je umístění v prostoru pouze atributem jednotlivých prvků

systemu, nikoli parametrem hrající roli při výpočtu.

Ze své podstaty jsou P systémy diskrétní a transport je realizován na úrovni jednotlivých objektů. Čas transportu, který je pro simulování reálných jevů zásadní, probíhá v předem daných časových jednotkách. Použití časovaných P systémů (Cavaliere a Zandron, 2006; Barbuti et al., 2008) umožňuje přiřadit pravidlům čas vykonání, který musí být násobkem základní časové jednotky.

Oba modely navazují na P systémy se symportem/antiportem (Păun a Păun, 2002) a generalizované P systémy (Verlan et al., 2008), neboť pravidla, která jsou použita, jsou pravidly komunikačními. Dochází pouze k přesunu objektů v rámci systému, nikoli k vytváření nových objektů. Nové objekty mohou do systému vstupovat pouze z vnějšího prostředí. Kromě přesunu v rámci systému mohou objekty měnit svou kvalitu ( $u \rightarrow v$ ), nikoli však svou kvantitu ( $u \rightarrow uu$  není povoleno).

Model *transportních P systému se signalizací, kapacitami a nosiči* simuluje transport individuálních objektů v síti. Jsou využívány principy časovaných pravidel (Cavaliere a Zandron, 2006), katalyzátorů a stavů synapsí (podobně jako u tkáňovitých P systémů se stavy komunikačních kanálů (Freund et al., 2004)). Inspirací při návrhu modelu byl pohyb vozidel v dopravní síti se signalizovanými křižovatkami. Každá z membrán v systému reprezentuje buď křižovátku nebo vstupní a výstupní bod systému. Mezi kteroukoli dvojicí membrán může existovat nejvýše jedna synapse v každém směru. Existence více synapsí v některém ze směrů by vyžadovala změnu definice modelu.

Synapse mohou nabývat právě jeden z předem definované množiny stavů, přičemž aplikace pravidel může být omezena existencí synapse v daném stavu. Ve všech příkladech použití zmíněných v této práci jsou však použity pouze dva stavy –  $s^+$  a  $s^-$ , reprezentující otevřený a zavřený komunikační kanál.

Synapse zde v podstatě plní roli promotora (Ionescu a Sburlan, 2004), protože jejich přítomnost (v některém ze stavů) může být podmínkou vykonání pravidla. Objekty v systému jsou rozlišeny na běžné objekty a katalyzátory. Katalyzátory v rámci systému nemění své množství ani pozici a umožňují aplikaci pravidel. Rozdíl mezi katalyzátorem a promotorem (synapsí) spočívá v tom, že existence jednoho katalyzátoru umožní aplikaci právě jednoho pravidla, kdežto existence promotora umožní aplikaci libovolného počtu pravidel.

Katalyzátory v navrženém modelu reprezentují kapacitu křižovatek – počet vozidel, která mohou křižovatekou za časovou jednotku projet, zatímco stavy synapsí

reprezentují průjezdnost křižovatkou, tedy signalizaci (světlou či jinou).

Nedeterminismus systému je omezen použitím pravděpodobností pro výběr pravidel, čímž model navazuje na práce (Obtulowicz a Păun, 2003; Pescini et al., 2006). Pravděpodobnost je použita na úrovni výběru pravidel, kdy každé pravidlo má zadánu pseudopravděpodobnost výběru. Pro účely simulování dopravy může být pseudopravděpodobnost určena na základě pozorovaných intenzit dopravy.

Model není navržen ke sledování cesty konkrétních objektů v síti. U objektů je evidována pouze jejich aktuální poloha a indikátor následující stanice, nikoli celá cesta. Výsledky výpočtu nelze použít k činění závěrů o délce či průběhu cesty, pouze ke sledování množství objektů, které se v daných časových okamžicích nacházely v jednotlivých částech systému. Výjimku tvoří situace, kdy systém obsahuje jediný objekt, jak bylo ukázáno v podkapitole 4.7 při simulaci náhodné procházky.

Inovativními rozšířeními modelu jsou tzv. události a visící synapse. Události jsou pravidly, jejichž aplikace je jednorázová a je vázána na konkrétní časový krok. Příkladem jejich použití je například nenadálé přerušení komunikace, odstranění objektů z konkrétní membrány nebo naopak vložení objektů do jedné nebo více membrán. Vzhledem k tomu, že tvar pravidla v navrženém modelu obsahuje indikátor membrány, v níž se pravidlo nachází, není nutné umisťovat události do konkrétních membrán, v nichž mají být vykonány, ale mohou být definovány na úrovni membránového systému.

Visící synapse bylo nutné zavést z důvodu, že navržený model kombinuje variabilní délku vykonání pravidel se změnou stavů synapsí. Visící synapse označují ty synapse, jejichž stav je právě měněn některým z visících pravidel. Stav visících synapsí nemůže být měněn. Toto omezení zaručuje, že synapse budou vždy právě v jednom stavu. Pokud by neexistovalo, mohlo by dojít k situaci, kdy během jednoho výpočetního kroku má být stav synapse změněn do několika stavů. Na zcela obecné úrovni to nemusí být problém – nedeterministicky by se vybralo jedno z pravidel, respektive pravidla by byla vykonána v náhodném pořadí. Pro účely modelování reálného transportu však bylo nutné zaručit, aby se stavy synapsí měnily v pevně daných časových intervalech a nedeterminismus byl odstraněn.

Ačkoli hlavním objektem zájmu této práce je využití P systémů pro simulace a nikoli výpočty, je poskytnut důkaz, že představený model je Turingovsky kompletní.

Pro model *transportních P systémů se signalizací, kapacitami a událostmi* je vy-

tvořen softwarový nástroj v jazyce **R**, který je spolu s návodem k použití přílohou této práce a je dostupný ze stránek [https://github.com/janoskaz/trans\\_p\\_systems](https://github.com/janoskaz/trans_p_systems).

Pomocí softwarového nástroje jsou vytvořeny simulace, které demonstrují použití navrženého modelu k řešení jak ryze teoretických problémů, tak i k simulaci reálných systémů.

Model je použit k simulaci náhodné procházky a tzv. *ruinování hráče*, které z ní vychází. Simulace ukazují, že navržený model je schopen simulovat náhodnou procházku a to jak v případě, že se jedná o tzv. férovou hru (pravděpodobnosti všech možných kroků jsou shodné), tak v případě neférové hry. P systémy jsou obzvláště vhodné pro simulování náhodné procházky s omezeným prostorem (pohlcující nebo odrážející stěny). Simulace *ruinování hráče* ukazují, že výsledky her simulovaných modelem odpovídají očekávaným hodnotám, vypočteným numericky. Simulace náhodné procházky a ruinování hráče nevyžadují použití stavů synapsí ani událostí.

Výhodou představeného modelu je nedeterminismus výpočtu. Pokud má být provedena simulace tisíce her, není nutné tisíckrát simulovat hru jednoho hráče; do počítačnické membrány je vloženo tisíc hráčů a je simulována jedna hra.

Další simulace ukazují možnost použití modelu pro modelování šíření invazivního druhu v rámci říční sítě a simulování šíření znečištění. Stav synapsí je zde použit k reprezentaci zamrznání hladiny vodních toků, katalyzátory reprezentují dostupné potravní zdroje a události mohou být použity k simulaci jednorázových zásahů do systému – ať už je to vyhubení organismu v určité části systému, nebo naopak vložení populace do jedné z membrán.

Poslední simulace se zabývá transportem vozidel v rámci části silniční sítě a he ukázkou modelování procesů, k nimž byl model původně navržen. Události zde slouží k modelování přejezdu vlaku, který způsobí dočasnou neprůjezdnost určité synapse. Stav synapsí jsou použity k řízení křižovatky a kapacita určuje počet vozidel, která mohou křižovatkou během jedné časové jednotky projet.

Prezentované simulace ukazují poměrně široké použití modelu. Ačkoli byl model navržen na základě transportu vozidel v silniční síti, paradoxně vidím větší použití při modelování jevů, spadajících do oblasti fyzické geografie (sedimentace látek, jejich šíření v říční síti) nebo biogeografie (populační biologie). Silou představených modelů je práce s jednotlivými objekty, narozdíl od práce s populací, popsanou sadou parametrů. Toto může být výhodou zejména při modelování vývoje málo početných systémů, například reliktních populací. Model lze rovněž použít pro matematické modelování, například náhodných procházek v grafu.

Druhý představený model se nazývá *transportní P systémy s nosiči* a k transportu objektů používá mobilních membrán. Blíže se podobá P systémům s nosiči (Martín-Vide et al., 2002b). *Transportní P systémy s nosiči* byly ve svém návrhu inspirovány transportem cestujících v prostředcích hromadné dopravy.

Rozdílným prvkem oproti *transportním P systémům* je způsob přepravy objektů mezi membránami, který neprobíhá samostatně, ale uvnitř nosiče – mobilní membrány, která je posílána z jedné membrány do druhé. Mobilní membrány mohou vznikat a zanikat a obsahují pravidla (tzn. nejsou to jen prázdné schránky). Oproti P systémům s nosiči (Martín-Vide et al., 2002b) se navržený model liší ve dvou ohledech. P systémy s nosiči pracují s hierarchickou strukturou membrán (*transportní P systémy s nosiči* pracují s membránami uspořádanými do sítě) a nosič je po přemístění do membrány vždy rozpuštěn. Transport je realizován vždy mezi dvěma membránami. V *transportních P systémech* mohou mobilní membrány existovat libovolně dlouho. Objekt může projít membránou uvnitř nosiče, aniž by přišel do styku s ostatními objekty, které se v membráně nacházejí. V P systémech s nosiči jsou po každém transportu mezi membránami objekty umístěny přímo do membrány, zatímco ve zde navrženém modelu zůstávají v nosiči, který může zamezit jejich interakci s obsahem membrány, v níž se nosič nachází.

Kapacita vozidel je reprezentována pomocí katalyzátorů, jejichž počet v mobilní membráně odpovídá maximálnímu počtu objektů, které mohou být přepravovány.

Jak ukazuje kapitola 4.9.3, přizpůsobení navrženého modelu tak, aby věrohodně simuloval reálnou dopravu vyžaduje komplikované rozšíření, které efektivně vede k téměř úplnému omezení nedeterminismu systému. Nedeterminismus, společně s paralelismem jsou však základními vlastnostmi P systémů. Navržený model se proto k modelování reálných dopravních systémů příliš nehodí. Přeprava cestujících a osobní doprava obecně je tématem, které je dobře prozkoumáno a znalosti problematiky umožňují sestavení detailních modelů (Hoogendoorn a Bovy, 2001a). P systémy se ukázaly vhodné spíše pro modelování systémů, které jsou inherentně nedeterministické. Osobní doprava je oproti tomu systémem, ve kterém kritickou roli hrají rozhodovací procesy. Ty nemohou být v rámci P systémů věrohodně reprezentovány a chování může být popsáno pouze sadou pravidel, kterým je určena pravděpodobnost či priority. Rozhodování na úrovni vyhodnocování aktuální situace je procesem, který lze mnohem lépe modelovat například multiagentními systémy.

Zde navržené modely jsou proto vhodnější pro reprezentaci procesů, při nichž nedochází k vědomému rozhodování a umožňují provádět simulace s nastavením růz-



ných parametrů. Dále jsou vhodné pro modelování procesů, o nichž máme omezené znalosti, nebo nejsou dostupná data, která by umožnila sestavení detailních modelů.

P systémy byly navrženy jako model, který v sobě skloubí nedeterminismus a maximální paralelismus. Proto je vhodné použít je k modelování jevů, které ze své podstaty mají tyto vlastnosti. Jevy, které jsou deterministické (a osobní doprava mezi ně z velké části patří), nevyužívají plně vlastností P systémů a expresivnost, kterou P systémy poskytují, nemusí být v důsledku dostatečným důvodem k tomu, aby byl tento model zvolen.

## 6 DISKUZE

Navržené modely transportu objektů v síti staví na základním modelu P systémů (Păun, 1998), které představují distribuovaný, nedeterministický a maximálně paralelní výpočetní model. Tyto charakteristiky platí i pro zde navržené modely. Při výpočtu je předpokládána existence globálních hodin s konstantním krokem. Jedná se tedy o výpočetní model, kdy výpočty v jednotlivých částech systému (membránách) probíhají nazávisle na sobě.

Výhodou tohoto modelu je možnost simulace velkého počtu procesů v rámci jednoho běhu simulace. Například v kapitole 4.8 je ukázáno, jak může být navržený model použit k simulaci stovek běhů simulace ruinování hráče současně. Při použití numerických modelů by bylo nutné simulovat jednotlivě  $n$  běhů programu a posléze agregovat výsledky, zatímco P systémy umožňují současnou simulaci  $n$  her v rámci jednoho výpočtu. Vzhledem k tomu, že při studiu transportu se zřídka zabýváme transportem jedné entity, představují P systémy obecně perspektivní výpočetní model.

Současně však P systémy neumožňují sledovat evoluci nebo pohyb jednotlivých objektů (kromě vzácné situace, kdy se v systému nachází jediný objekt). Mezi jednotlivými objekty není rozlišováno, z pohledu modelu se jedná o totožné objekty. Pokud se například v jedné membráně (křižovatce) nachází objekty reprezentující vozidla, jež přišly ze dvou různých směrů, jedná se o tytéž objekty a nelze rozlišit, který objekt přišel ze které membrány.

Vhodným použitím navrženého modelu *transportních P systémů* je například modelování disperze, kdy předpokládáme velké množství látky v počátečním místě a sledujeme, jak se v průběhu výpočtu mění její množství v dalších částech systému. Podobnou úlohou je modelování vývoje populací, čímž se zabývali například (Besozzi et al., 2008). Zde však představené modely transportu naráží na problém s reprezentací růstu populací. Oba modely používají pravidla, která popisují změnu polohy a kvality, ale nikoli kvantity. Pro modelování dynamiky růstu populace je však nutné použít pravidla, která mění počet objektů v systému (tvar pravidla  $u \rightarrow uu$ ). Ačkoli není problém rozšířit model *transportních P systémů* tak, aby pravá i levá strana pravidla mohly obsahovat více než jeden běžný objekt (vedle běžného objektu obsahují katalyzátor a synapsi), jednalo by se již o jiný model s odlišnými vlastnostmi. *Transportní P systémy* umožňují kromě přemístování objektu měnit i jeho kvalitu – například mohou existovat objekty  $a'$ ,  $a''$ , ...,  $a^x$ , z nichž každý reprezentuje jeden stav modelovaného objektu (například přemístované chemické látky, nebo fázi života or-

ganismu). Změna počtu objektů v systému je možná pouze skrz komunikaci s vnějším prostředím.

Zatímco paralelismus P systémů je jednoznačně výhodnou vlastností, umožňující efektivní modelování velkého počtu operací současně, nedeterminismus naopak představuje problém při modelování reálných dopravních systémů. Dopravní systémy nejsou nedeterministické, zpravidla jsou dány počátky a cíle přepravovaných objektů a zkoumá se detailně jejich cesta. P systémy jsou sice schopny takové chování popsat, jak ukazuje kapitola 4.9.3, není to však vhodné použití modelu – takto upravený model omezuje nedeterminismus na holé minimum.

Existuje řada variant P systémů, kde je nedeterminismus částečně omezen, ať již se jedná o použití pravděpodobnosti při výběru pravidel (Obtulowicz a Păun, 2003), stanovení priority pravidel, tichá pravidla (Besozzi et al., 2008) nebo jsou to P systémy, které pracují asynchroně (Cavaliere, 2004) nebo sekvenčně (Dang a Ibarra, 2004). Zde je nedeterminismus omezen jednak použitím cílových indikátorů u přepravovaných objektů, jednak přítomností katalyzátorů a synapsí v daném stavu a konečně i pseudopravděpodobností výběru pravidla. Ani tato omezení nedeterminismu však nejsou dostatečná pro věrohodnou simulaci hromadné dopravy. Problémem je například existence předem daných linek vozidel. Klasická varianta P systémů předpokládá náhodný výběr pravidel. Pokud tedy existuje pravidlo, které posílá objekt nebo mobilní membránu do některé ze sousedních membrán, měla by být sousední membrána vybrána náhodně. Reálná doprava však takto nefunguje a přepravní cesty jsou předem známé a neměnné.

Pokud je modelována přeprava jednotlivých objektů, je možné ji řídit na základě pravděpodobností použití určité cesty, což je princip, který používají *transportní P systémy se signalizací, kapacitou a událostmi* (viz kap. 4.2). Nejsou sledovány individuální cesty vozidel, ale jejich počet v jednotlivých částech systému. V případě přepravy dopravními prostředky však není možné umožnit vozidlům, aby náhodně cestovala mezi jednotlivými stanicemi; musí být pevně daná linka, po níž se bude vozidlo pohybovat.

Obdobný problém představuje rozhodování cestujících o výběru cesty. V reálu cestující přesně ví, která stanice je jeho cílová a jakou cestou se do ní chce dostat. P systémy předpokládají, že objekty jsou jednak zaměnitelné (všichni cestující nacházející se ve vozidle/stanici jsou reprezentováni jako totožné objekty) a jejich rozhodování (*nastoupím/setrvám/vystoupím*) je řízeno náhodným výběrem pravidla z předem dané množiny. V rámci navržených *transportních P systémů s nosiči* je představeno rozší-

ření modelu, kde každému cestujícímu je přiřazen seznam stanic v pořadí, ve kterém je cestující navštíví. Takováto reprezentace umožňuje simulaci přepravy cestujících mezi libovolnou dvojicí stanic, efektivně však vede k odstranění nedeterminismu systému.

Lze si představit obecný systém, kdy se sítí pohybují vozidla dle principu náhodné procházky, do nich nastupují a v náhodných částech systému z nich vystupují přepravované objekty. Takový systém lze pomocí *transportních P systémů s nosiči* modelovat, neblíží se však reálné dopravě.

Jsou-li předem dány jak jízdní řády vozidel, tak jejich cesty a má-li každý objekt určen svou cestu, je model redukován na de-facto multiagentní systém, v němž nejsou pravidla aplikována nedeterministicky, ale podle předem daného klíče, v němž neexistuje žádná variabilita. Jediným zdrojem nedeterminismu může být časová distribuce dopravní poptávky (počet cestujících, kteří přicházejí do systémů), jejich cílové stanice a frekvence a kapacita vozidel. Všechny tyto proměnné je však obtížné reprezentovat v P systémech. Časy odjezdů vlaků, počty cestujících a jejich cílové stanice musí být generovány podle klíče, který nelze v rámci zde představených modelů zachytit a musí být řízen procesem mimo rámec navrženého modelu.

Ani po vyřešení těchto problémů by však nebylo možné modelovat situace, jako jsou například změny tras v závislosti na aktuální situaci (intenzitě dopravy).

P systémy jsou tedy vhodné pro modelování transportu, který má spíše mechanickou povahu (například unášení látek vodním proudem) a nejsou vhodné pro modelování procesů, které jsou deterministické a obsahují rozhodovací procesy.

Další obecnou výhodou P systémů je diskrétnost modelovaných objektů. Ve studiích, zabývajících se například populační dynamikou, se mnohdy setkáváme s problémem modelování reliktních, málo početných populací (Barbuti et al., 2008; Colomer et al., 2011). Numerické modely, založené na systému diferenciálních rovnic, nepracují s jednotlivci ale celou populaci vyjadřují pomocí jedné kvantity (Freedman, 1982). P systémy, pracující s jednotlivými objekty, jsou lépe schopny modelovat málo početné populace, kdy kolaps celé populace může záviset na přežívání několika málo jedinců. Obdobně modely transportu vozidel v síti, založené na kinetických rovnicích, popisují dopravní proud jako vektor čísel ale nerozlišují mezi jednotlivými vozidly (Hoogendoorn a Bovy, 2001a). V případě modelování transportu nízkých počtů přepravovaných entit (např. disperze organismů, náhodná procházka v grafu apod.) jsou P systémy vhodným výpočetním modelem.

Při modelování dopravy je nutné pracovat s časem. P systémy umožňují přiřadit každému pravidlu čas vykonávání, který však musí být násobkem základního časového intervalu. Fakt, že existují globální hodiny a čas ve všech částech systému plyne stejně není limitující faktor, naopak odpovídá modelované skutečnosti. Oba zde navržené modely transportu se zabývají především chováním objektů v membránách, reprezentujících místa v síti a nikoli chováním na komunikacích, které jsou zde reprezentovány pomocí synapsí, spojujících membrány. Transport mezi dvěma membránami je popsán pomocí pravidla, které má přiřazenou časovou konstantu. Není možné zachytit ani chování jednotlivých objektů během transportu mezi membránami (zrychlování, zpomalování, předjížděcí manévry), ani rozdílnou délku cesty. Je možné rozlišovat mezi různými typy vozidel a transport každého typu popsat pravidlem s rozdílnou časovou konstantou. Například transport objektu *car*, reprezentujícího osobní automobil může trvat kratší dobu než transport objektu *bus*, reprezentujícího autobus či nákladní automobil. Dále je možné stanovit rozdílný čas vykonání pravidla v obou směrech cesty a reprezentovat tak rozdílnou délku cesty v obou směrech.

Výše napsané znamená, že P systémy obecně a zde představené modely konkrétně nejsou použitelné pro modelování detailních interakcí vozidel (či jiných přepravovaných objektů).

Ačkoli se tato práce zabývá modelováním prostorových jevů, reprezentovat v P systémech vzdálenost je obtížné. Inspirací pro P systémy byla živá buňka, v níž nezáleží na vzdálenosti objektů ale na jejich koncentraci (Păun, 1998). Předpokládá se, že vše, co se nachází v jedné membráně je dostatečně blízké na to, aby spolu mohlo interagovat, zatímco vše, co je v rozdílných membránách, je příliš vzdálené, aby spolu mohlo interagovat. V rámci objektů, nacházejících se v membráně, neexistuje pořadí, pouze koncentrace.

Snaha definovat v rámci P systémů vzdálenost je proto v protikladu s myšlenkovým rámcem, z něhož tento model vzešel. Tato práce neměla ambice rozšířit model P systémů o práci se vzdáleností nebo obecně prostorem, pouze v rámci již publikovaných rozšíření našla principy, které umožňují s prostorem nepřímo pracovat.

Zde představené modely uvažují, že membrány tvořící síť představují entity reálného světa, proto jejich atributem může být i poloha v rámci zvoleného souřadného systému. Jejich vzdálenost nehraje při výpočtu roli, je však možné definovat mezi nimi sousedství a to skrz vytvoření synapse. Synapse mohou existovat mezi kteroukoli dvojicí membrán v obou směrech. Omezením obou modelů je možnost existence pouze jedné synapse v každém směru, protože synapse je definována jako dvojice  $(i, j)$ , kde

$i$  a  $j$  jsou popisky membrán. V případě, že by existovalo více synapsí mezi dvojicí membrán, musela by každá synapse být individuálně identifikovatelná. Při aplikaci pravidel by dále muselo být stanoveno, jaké jsou podmínky aplikace pravidel, která mění stav synapse (mění se stav jedné konkrétní synapse? náhodně vybrané synapse, která splňuje podmínky? všech synapsí, které splňují podmínky?). V principu se nejedná o problém, pro účely této práce však stačí pracovat s maximálně jednou synapsí v každém směru.

Synapse jsou pouze mentální spojnicí membrán, nereprezentují existující fyzické objekty (silnice, letecké trasy apod.). Synapse dále neobsahují informace o vzdálenosti membrán, které spojují, pouze nesou informaci o tom, že mezi oběma membránami lze přímo transportovat objekty.

Vzdálenost membrán může být vyjádřena dvěma způsoby – jako pravděpodobnost či jako trvání pravidla. Pravděpodobnost je méně častým způsobem vyjádření a lze použít tam, kde pracujeme s pravidly s jednotkovým časem vykonání. Například můžeme modelovat pohyb jedinců v rámci systému, kde jsou všechny membrány navzájem spojeny synapsími a každý jedinec se během jednoho časového kroku přesune z jedné membrány do druhé. Přitom je větší pravděpodobnost, že se přesune do membrány bližší, než že se přesune do membrány vzdálenější. V tomto případě můžeme definovat pravděpodobnost  $p_{ij}$  pro každé pravidlo, popisující transport z membrány  $i$  do membrány  $j$  jako funkci vzdálenosti (například  $1/d$ , kde  $d$  je vzdálenost membrán).

Častější způsob reprezentace času však bude skrz dobu vykonání pravidla, tedy pomocí času. Transport mezi objekty, které jsou vzdálenější, zabere více času. Tudíž vzdálenost mezi dvojicí membrán lze reprezentovat jako čas  $t$ , který zabere vykonání pravidla, přemístujícího objekt  $u$  z membrány  $i$  do membrány  $j$ . V nejjednodušším případě je rychlost pohybu všech objektů konstantní v celém systému a pak čas vykonání transportu je roven vzdálenosti počátku a cíle. Můžeme se však setkat se systémy, kdy transport více závisí na čase, než na vzdálenosti. Například transport v říční síti bude probíhat diametrálně odlišně po a proti proudu.

Použití času vykonání pravidla při modelování transportu je tedy jeví jako správný způsob reprezentace vzdálenosti mezi objekty .

## 7 ZÁVĚR

P systémy představují výpočetní model, inspirovaný strukturou a fungováním živé buňky. Základem P systémů je membránová struktura, v jejíchž odděleních jsou obsaženy objekty a pravidla, která jimi manipulují. Výpočet probíhá aplikací pravidel na množinu objektů, přičemž dochází ke změnám kvality i kvantity objektů a k jejich přemístování v rámci membránové struktury.

Tato práce se zabývá možností využití P systémů pro modelování transportu. Téma transportu v membránových systémech bylo široce zkoumáno, téměř výhradně však z pohledu biochemického. Transport v geografickém prostoru je odlišný zejména proto, že se neděje v hierarchické struktuře oddělení (kterou klasické P systémy představují), ale v rovině (případně ve vyšších rozměrech). Práce navazuje na řadu publikací, které se transportem v P systémech zabývaly a představuje dva modely transportu, jež byly sestaveny na základě inspirace fungováním dopravy osob a zboží. Oba modely byly sestaveny za účelem provádění simulací a nikoli výpočtů.

*Transportní P systém se signalizací, kapacitami a událostmi* modeluje transport objektů v síti, jejíž uzly tvoří membrány. Membrány mohou obsahovat katalyzátory, simulující kapacitu dopravní stavby (křižovatky, nebo jiného prvku dopravní infrastruktury) a synapse spojující membrány mohou nabývat několika stavů, čímž je reprezentována průjezdnost komunikací. Inovativním prvkem je použití událostí, které mají podobu pravidel, která jsou jednorázově aplikována v daném časovém kroku. Pomocí událostí lze simulovat například přerušení komunikace nebo náhlé zvýšení/snížení počtu přepravovaných objektů v některé z částí systému. Je dokázána Turingovská kompletnost navrženého modelu.

*Transportní P systém s nosiči* modeluje transport objektů pomocí vozidel. Podobá se modelu *P systémů s nosiči*, v některých stěžejních bodech se od něj však liší. Základem modelu je opět síť membrán, spojených synapsami. Transport objektů mezi membránami však neprobíhá přímo, ale pomocí mobilních membrán (nosičů), které jsou mezi jednotlivými membránami v síti přemístovány. Mobilní membrány opět obsahují katalyzátory, reprezentující kapacitu nosičů.

Pro *transportní P systém se signalizací, kapacitami a událostmi* je vytvořen simulátor v jazyce **R**, který je přílohou této práce a umožňuje návrh libovolného P systému na základě představeného modelu a jeho simulaci.

V simulátoru jsou provedeny simulace modelových systémů, na kterých je ukázána versatilita modelu. Je provedena simulace náhodné procházky, z ní odvozené hry ruinování hráče, simulace šíření invazivního druhu říční sítí, simulace šíření znečištění z

bodového zdroje v říční síti a simulace přepravy vozidel v jednoduché síti s křižovatkou a železničním přejezdem.

Ačkoli byly oba modely navrženy za účelem simulace přepravy osob a nákladu, větší uplatnění dle mého názoru najdou při simulaci mechanického transportu; například sedimentace, šíření znečištění apod. Osobní doprava je systémem, který je silně deterministický. Nutnost reprezentovat parametry systému, jakými jsou jízdni řády, trasy vozidel a osob, měnící se dopravní poptávka apod. jde proti nedeterminismu P systémů, a vhodnějším nástrojem modelování mohou být například multiagentní systémy, které umožňují modelovat rozhodovací procesy, jež ze své podstaty v P systémech nemohou být vyjádřeny.



# POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

- ALHAZOV, A., ISHDORJ, Ts.-O. Membrane operations in P systems with active membranes. In PĂUN, Gh., RISCOS-NÚÑEZ, A., ROMERO-JIMÉNEZ, A., SANCHO-CAPARRINI, F. (Ed.) *Second Brainstorming Week on Membrane Computing Sevilla, Spain, February 2-7 2004*, s. 37–44, Sevilla, Spain, February 2-7 2004. Dostupné z: <http://www.gcn.us.es/2BWMC/Volumen.htm>.
- ALHAZOV, A., PAN, L., PĂUN, Gh. Trading polarizations for labels in P systems with active membranes. *Acta Informatica*, 41, 2-3, s. 111–144, December 2004. Dostupné z: <http://dx.doi.org/10.1007/s00236-004-0153-z>.
- AMAN, B., CIOBANU, G. Describing the immune system using enhanced mobile membranes. *Electronic Notes in Theoretical Computer Science*, 194, 3, s. 5 – 18, 2008. ISSN 1571-0661. doi: <http://dx.doi.org/10.1016/j.entcs.2007.12.003>. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S1571066108000224>.
- BAEE, S., ESHGHI, F., HASHEMI, S. M., MOIENFAR, R. Passenger boarding/alighting management in urban rail transportation. In *Proceedings of the 2012 Joint Rail Conference*, 2012.
- BARBUTI, R., MAGGIOLO-SCHETTINI, A., MILAZZO, P., TESEI, L. Timed P automata. In CIOBANU, G. (Ed.) *Second International Meeting on Membrane Computing and Biologically Inspired Process Calculi*, s. 53–67, 2008.
- BARBUTI, R., MAGGIOLO-SCHETTINI, A., MILAZZO, P., TINI, S. P systems with transport and diffusion membrane channels. *Fundam. Inf.*, 93, 1-3, s. 17–31, January 2009. ISSN 0169-2968. Dostupné z: <http://dl.acm.org/citation.cfm?id=2364580.2364583>.
- BARBUTI, R., MAGGIOLO-SCHETTINI, A., MILAZZO, P., PARDINI, G., TESEI, L. Spatial p systems. *Natural Computing*, 10, 1, s. 3–16, 2011. ISSN 1572-9796. doi: [10.1007/s11047-010-9187-z](http://dx.doi.org/10.1007/s11047-010-9187-z). Dostupné z: <http://dx.doi.org/10.1007/s11047-010-9187-z>.
- BARBUTI, R., MAGGIOLO-SCHETTINI, A., MILAZZO, P., PARDINI, G. Simulation of spatial p system models. *Theoretical Computer Science*, 529, s. 11 – 45, 2014. ISSN 0304-3975. doi: <http://dx.doi.org/10.1016/j.tcs.2013.08.002>. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S030439751300580X>.

- BAZZAN, A. L. C., KLUGL, F. *Multi-Agent Systems for Traffic and Transportation Engineering*. Hershey, PA : Information Science Reference - Imprint of: IGI Publishing, 2009. ISBN 1605662267, 9781605662268.
- BERNARDINI, F., GHEORGHE, M., KRASNOGOR, N. Population P systems and quorum sensing in bacteria. *Theoretical Computer Science*, 2006.
- BERNARDINI, F., GHEORGHE, M. Population P systems. *Journal of Universal Computer Science*, 10, 5, s. 509–539, May 2004.
- BERNARDINI, F., MANCA, V. Dynamical aspects of P systems. *BioSystems*, 70, 2, s. 85–93, July 2003.
- BESOZZI, D., CAZZANIGA, P., PESCHINI, D., MAURI, G. Seasonal variance in p system models for metapopulations. *Progress in Natural Science*, 17, 4, s. 392–400, 2007. doi: 10.1080/10020070708541015.
- BESOZZI, D., CAZZANIGA, P., PESCHINI, D., MAURI, G. Modelling metapopulations with stochastic membrane systems. *Biosystems*, 91, 3, s. 499 – 514, 2008. ISSN 0303-2647. doi: <http://dx.doi.org/10.1016/j.biosystems.2006.12.011>. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0303264707001165>. P-Systems Applications to Systems Biology.
- BJÖRKLUND, G. M., ÅBERG, L. Driver behaviour in intersections: Formal and informal traffic rules. *Transportation Research Part F: Traffic Psychology and Behaviour*, 8, 3, s. 239 – 253, 2005. ISSN 1369-8478. doi: <http://dx.doi.org/10.1016/j.trf.2005.04.006>. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S1369847805000252>.
- BLUMEN, A., KLAFTER, J., ZUMOFEN, G. *Models for Reaction Dynamics in Glasses*, s. 199–265. Springer Netherlands, Dordrecht, 1986. doi: 10.1007/978-94-009-4650-7\_5. Dostupné z: [http://dx.doi.org/10.1007/978-94-009-4650-7\\_5](http://dx.doi.org/10.1007/978-94-009-4650-7_5). ISBN 978-94-009-4650-7.
- BOTTONI, P., MARTÍN-VIDE, C., PĂUN, G., ROZENBERG, G. Membrane systems with promoters/inhibitors. *Acta Informatica*, 38, 10, s. 695–720, September 2002.
- CACCIAGRANO, D., CORRADINI, F., MERELLI, E., TESEI, L. Multiscale bone remodelling with spatial P systems. In *Proceedings Fourth Workshop on Membrane*

- Computing and Biologically Inspired Process Calculi, MeCBIC 2010, Jena, Germany, 23 August 2010.*, s. 70–84, 2010. doi: 10.4204/EPTCS.40.6. Dostupné z: <http://dx.doi.org/10.4204/EPTCS.40.6>.
- CAPRA, F., LUISI, P. L. (Ed.). *The Systems View of Life: A Unifying Vision*. Cambridge University Press, 2014. ISBN 978-1316616437.
- CARDONA, M., COLOMER, M. A., MARGALIDA, A., PÉREZ-HURTADO, I., PÉREZ-JIMÉNEZ, M. J., SANUY, D. *A P System Based Model of an Ecosystem of Some Scavenger Birds*, s. 182–195. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. doi: 10.1007/978-3-642-11467-0\_14. Dostupné z: [http://dx.doi.org/10.1007/978-3-642-11467-0\\_14](http://dx.doi.org/10.1007/978-3-642-11467-0_14). ISBN 978-3-642-11467-0.
- CARDONA, M., COLOMER, M. A., MARGALIDA, A., PALAU, A., PÉREZ-HURTADO, I., PÉREZ-JIMÉNEZ, M. J., SANUY, D. A computational modeling for real ecosystems based on p systems. *Natural Computing*, 10, 1, s. 39–53, 2011. ISSN 1572-9796. doi: 10.1007/s11047-010-9191-3. Dostupné z: <http://dx.doi.org/10.1007/s11047-010-9191-3>.
- CAVALIERE, M. Towards asynchronous P systems. In *Pre-proceedings of the Fifth Workshop on Membrane Computing (WMC5), Milano, Italy, June 2004*, s. 161–173, Milano, Italy, June 2004.
- CAVALIERE, M., ZANDRON, C. Time-driven computations in P systems. In GUTIÉRREZ-NARANJO, M. A., PĂUN, G., RISCOS-NÚÑEZ, A., ROMERO-CAMPERO, F. J. (Ed.) *Fourth Brainstorming Week on Membrane Computing, Sevilla, January 30 - February 3, 2006. Volume I*, s. 133–144. Fénix Editora, 2006.
- CIOBANU, G., PĂUN, G., PÉREZ-JIMÉNEZ, M. J. *Applications of Membrane Computing*. Springer-Verlag, 2006. ISBN 978-3-540-25017-3.
- CODLING, E. A., PLANK, M. J., BENHAMOU, S. Random walk models in biology. *Journal of the Royal Society Interface*, 5, 25, s. 813–834, August 2008. doi: 10.1098/rsif.2008.0014.
- COLOMER, M. A., LAVÍN, S., MARCO, I., MARGALIDA, A., PÉREZ-HURTADO, I., PÉREZ-JIMÉNEZ, M. J., SANUY, D., SERRANO, E., VALENCIA-CABRERA, L. *Modeling Population Growth of Pyrenean Chamois (*Rupicapra p. pyrenaica*) by Using P-Systems*, s. 144–159. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. doi: 10.1007/978-3-642-18123-8\_13. Dostupné z: [http://dx.doi.org/10.1007/978-3-642-18123-8\\_13](http://dx.doi.org/10.1007/978-3-642-18123-8_13). ISBN 978-3-642-18123-8.

- COLOMER, M. C. M. A., PÉREZ-JIMÉNEZ, M. J., SANUY, D., MARGALIDA, A. Modeling ecosystems using P systems: The bearded vulture, a case study. In CORNE, D. W., FRISCO, P., PĂUN, G., ROZENBERG, G., SALOMAA, A. (Ed.) *Membrane Computing: 9th International Workshop, 5391 / Lecture Notes in Computer Science*, s. 137–156, 2009.
- CSUHAIJ-VARJÚ, E., GHEORGHE, M., STANNETT, M., VASZIL, G. Spatially localised membrane systems. *Fundamenta Informaticae*, 138, 1-2, s. 193–205, 2015.
- DAAMEN, W. *Modelling Passenger Flows in Public Transport Facilities*. PhD thesis, Technical University of Delft, 2004.
- DANG, Z., IBARRA, O. H. On P systems operating in sequential and limited parallel modes. In *Workshop on Descriptive Complexity of Formal Systems*, London-Ontario, 2004.
- DAVIDRAJUH, R., LIN, B. Exploring airport traffic capability using petri net based model. *Expert Systems with Applications*, 38, 9, s. 10923 – 10931, 2011. ISSN 0957-4174. doi: <http://dx.doi.org/10.1016/j.eswa.2011.02.134>. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0957417411003551>.
- AMOR, M. A. M. *Accelerating Membrane Systems Simulators using High Performance Computing with GPU*. PhD thesis, School of Computer Engineering, University of Seville, Seville, Spain, Seville, Spain, 2014.
- DVORSKÝ, J., JANOŠKA, Z., VOJÁČEK, L. P systems for traffic flow simulation. In CORTESI, A., CHAKI, N., SAEED, K., WIERZCHON, S. T. (Ed.) *CISIM, 7564 / Lecture Notes in Computer Science*, s. 405–415. Springer, 2012. Dostupné z: <http://dblp.uni-trier.de/db/conf/cisim/cisim2012.html#DvorskyJV12>. ISBN 978-3-642-33259-3.
- ENCISO, J., VARGAS, J., MARTÍNEZ, P. Modeling and simulation of passenger traffic in a national airport. *CoRR*, abs/1604.02704, 2016. Dostupné z: <http://arxiv.org/abs/1604.02704>.
- EPSTEIN, R. A. (Ed.). *The Theory of Gambling and Statistical Logic (Second Edition)*. Boston : Academic Press, second edition edition, 2010. doi: <http://dx.doi.org/10.1016/B978-0-12-374940-6.00016-4>. Dostupné z: <http://www.sciencedirect.com/science/article/pii/B9780123749406000164>. ISBN 978-0-12-374940-6.

- FELLER, W. *An Introduction to Probability Theory and Its Applications*. 1. Wiley, January 1968. Dostupné z: <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20&path=ASIN/0471257087>. ISBN 0471257087.
- FREEDMAN, H. Deterministic mathematical models in population ecology. *Canadian Journal of Statistics*, 10, 4, s. 315–315, 1982. ISSN 1708-945X. doi: 10.2307/3556198. Dostupné z: <http://dx.doi.org/10.2307/3556198>.
- FREUND, R. *Energy-Controlled P Systems*, s. 247–260. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. doi: 10.1007/3-540-36490-0\_16. Dostupné z: [http://dx.doi.org/10.1007/3-540-36490-0\\_16](http://dx.doi.org/10.1007/3-540-36490-0_16). ISBN 978-3-540-36490-0.
- FREUND, R. Asynchronous P systems. In *Pre-proceedings of the Fifth Workshop on Membrane Computing (WMC5), Milano, Italy, June 2004*, s. 12–28, Milano, Italy, June 2004.
- FREUND, R. *Asynchronous P Systems and P Systems Working in the Sequential Mode*, s. 36–62. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. doi: 10.1007/978-3-540-31837-8\_3. Dostupné z: [http://dx.doi.org/10.1007/978-3-540-31837-8\\_3](http://dx.doi.org/10.1007/978-3-540-31837-8_3). ISBN 978-3-540-31837-8.
- FREUND, R., PĂUN, G., PÉREZ-JIMÉNEZ, M. J. Tissue P systems with channel states. *Theoretical Computer Science*, 2004. In press.
- GEORGIOU, A., GHEORGHE, M. Generative devices used in graphics. In ALHAZOV, A., MARTÍN-VIDE, C., PĂUN, G. (Ed.) *Preproceedings of the Workshop on Membrane Computing*, s. 266–272, Tarragona, July 17-22 2003.
- GUIZZI, G., MURINO, T., ROMANO, E. A discrete event simulation to model passenger flow in the airport terminal. In *Proceedings of the 11th WSEAS International Conference on Mathematical Methods and Computational Techniques in Electrical Engineering*, MACTEE'09, s. 427–434, Stevens Point, Wisconsin, USA, 2009. World Scientific and Engineering Academy and Society (WSEAS). Dostupné z: <http://dl.acm.org/citation.cfm?id=1949006.1949078>. ISBN 978-960-474-124-3.
- HIDAS, P. Modelling lane changing and merging in microscopic traffic simulation. *Transportation Research Part C: Emerging Technologies*, 10, 5–6, s. 351 – 371, 2002. ISSN 0968-090X. doi: [http://dx.doi.org/10.1016/S0968-090X\(02\)00026-8](http://dx.doi.org/10.1016/S0968-090X(02)00026-8). Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0968090X02000268>.

- HOOGENDOORN, S. P., BOVY, P. H. State-of-the-art of vehicular traffic flow modelling. *Proc. of the Institution of Mechanical Engineers Part I Journal of Systems and Control Engineering*, 215, s. 283–303, 2001a.
- HOOGENDOORN, S. P., BOVY, P. H. Platoon-based multiclass modeling of multilane traffic flow. *Networks and Spatial Economics*, 1, 1, s. 137–166, 2001b. ISSN 1572-9427. doi: 10.1023/A:1011533228599. Dostupné z: <http://dx.doi.org/10.1023/A:1011533228599>.
- IONESCU, M., SBURLAN, D. On P systems with promoters/inhibitors. Technical Report 01/2004, Dept. of Computer Sciences and Artificial Intelligence, Univ. of Sevilla, 2004.
- JANOŠKA, Z., DVORSKÝ, J. P system based model of passenger flow in public transportation systems: a case study of prague metro. In SNÁŠEL, V., RICHTA, K., POKORNÝ, J. (Ed.) *DATESO, 971 / CEUR Workshop Proceedings*, s. 59–69. CEUR-WS.org, 2013. Dostupné z: <http://dblp.uni-trier.de/db/conf/dateso/dateso2013.html#JanoskaD13>.
- KESTING, A., TREIBER, M. Calibrating car-following models by using trajectory data: Methodological study. *Transportation Research Record: Journal of the Transportation Research Board*, , 2088, s. 148–156, 2008.
- KITAMURA, R., CHEN, C., PENDYALA, R. M., NARAYANAN, R. Microsimulation of daily activity-travel patterns for travel demand forecasting. *Transportation*, 27, 1, s. 25–51, 2000. ISSN 1572-9435. doi: 10.1023/A:1005259324588. Dostupné z: <http://dx.doi.org/10.1023/A:1005259324588>.
- KRISHNA, S. N., PĂUN, G. P systems with mobile membranes. *Natural Computing*, 4, 3, s. 255–274, 2005. ISSN 1572-9796. doi: 10.1007/s11047-005-3771-7. Dostupné z: <http://dx.doi.org/10.1007/s11047-005-3771-7>.
- LI, W., ZHU, W. A dynamic simulation model of passenger flow distribution on schedule-based rail transit networks with train delays. *Journal of Traffic and Transportation Engineering (English Edition)*, 3, 4, s. 364 – 373, 2016. ISSN 2095-7564. doi: <http://dx.doi.org/10.1016/j.jtte.2015.09.009>. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S2095756415305833>.
- LIGHTHILL, M. J., WHITHAM, G. B. On kinematic waves. ii. a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 229, 1178, s. 317–345,

1955. ISSN 0080-4630. doi: 10.1098/rspa.1955.0089. Dostupné z: <http://rspa.royalsocietypublishing.org/content/229/1178/317>.
- MADHU, M. Probabilistic rewriting P systems. *International Journal of Foundations of Computer Science*, 14, 1, s. 157–166, February 2003.
- MADHU, M., KRITHIVASAN, K. P systems with dynamic membrane polarization. *Romanian Journal of Information Science and Technology*, 4, 1-2, s. 135–154, 2001.
- MAERIVOET, S., MOOR, B. D. Cellular automata models of road traffic. *Physics Report*, 419, 1, s. 1–64, 2005.
- MANCA, V., BIANCO, L., FONTANA, F. Evolution and oscillation in P systems: Applications to biological phenomena. In MAURI, G., PĂUN, G., PÉREZ-JIMÉNEZ, M. J., ROZENBERG, G., SALOMAA, A. (Ed.) *Membrane Computing, International Workshop, WMC5, Milano, Italy, 2004, Selected Papers (G. Mauri, Gh. Paun, M.J. Perez-Jimenez, G. Rozenberg, A. Salomaa, eds.)*, LNCS, Springer-Verlag, Berlin, 2005, to appear, 2010.
- MARTÍN-VIDE, C., PAZOS, J., PĂUN, G., RODRÍGUEZ-PATÓN, A. *A New Class of Symbolic Abstract Neural Nets: Tissue P Systems*, s. 290–299. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002a. doi: 10.1007/3-540-45655-4\_32. Dostupné z: [http://dx.doi.org/10.1007/3-540-45655-4\\_32](http://dx.doi.org/10.1007/3-540-45655-4_32). ISBN 978-3-540-45655-1.
- MARTÍN-VIDE, C., PĂUN, G., ROZENBERG, G. Membrane systems with carriers. *Theoretical Computer Science*, 270, 1, s. 779 – 796, 2002b. ISSN 0304-3975. doi: [http://dx.doi.org/10.1016/S0304-3975\(01\)00117-7](http://dx.doi.org/10.1016/S0304-3975(01)00117-7). Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0304397501001177>.
- MINSKY, M. L. *Computation: Finite and Infinite Machines*. Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 1967. ISBN 0-13-165563-9.
- NAGEL, K. Particle hopping models and traffic flow theory. *Phys. Rev. E*, 53, s. 4655–4672, May 1996. doi: 10.1103/PhysRevE.53.4655. Dostupné z: <http://link.aps.org/doi/10.1103/PhysRevE.53.4655>.
- NELSON, P. A kinetic model of vehicular traffic and its associated bimodal equilibrium solutions. *Transport Theory and Statistical Physics*, 24, 1-3, s. 383–409, 1995. doi: 10.1080/00411459508205136. Dostupné z: <http://dx.doi.org/10.1080/00411459508205136>.
- NG, T., WRIGHT, M. et al. Introducing the moniac: an early and innovative economic model. *Reserve Bank of New Zealand Bulletin*, 70, 2007.

- OBTULOWICZ, A., PĂUN, G. (In search of) Probabilistic P systems. *BioSystems*, 70, 2, s. 107–121, July 2003.
- PAN, L., ISHDORJ, T.-O. P systems with active membranes and separation rules. *Journal of Universal Computer Science*, 10, 5, s. 630–649, may 2004.
- PAPAGEORGIOU, M. Some remarks on macroscopic traffic flow modelling. *Transportation Research Part A: Policy and Practice*, 32, 5, s. 323 – 329, 1998. ISSN 0965-8564. doi: [http://dx.doi.org/10.1016/S0965-8564\(97\)00048-7](http://dx.doi.org/10.1016/S0965-8564(97)00048-7). Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0965856497000487>.
- PĂUN, A., PĂUN, G. The power of communication: P systems with symport/antiport. *New Generation Computing*, 20, 3, s. 295–305, 2002. ISSN 1882-7055. doi: 10.1007/BF03037362. Dostupné z: <http://dx.doi.org/10.1007/BF03037362>.
- PĂUN, G. Computing with membranes. Technical Report 208, Turku Center for Computer Science-TUCS, 1998. ([www.tucs.fi](http://www.tucs.fi)).
- PĂUN, G. P systems with active membranes: Attacking np complete problems. *JOURNAL OF AUTOMATA, LANGUAGES AND COMBINATORICS*, 6, s. 75–90, 1999.
- PĂUN, G. Computing with membranes. *J. Comput. Syst. Sci.*, 61, 1, s. 108–143, August 2000. ISSN 0022-0000. doi: 10.1006/jcss.1999.1693. Dostupné z: <http://dx.doi.org/10.1006/jcss.1999.1693>.
- PĂUN, G. A quick introduction to membrane computing. *The Journal of Logic and Algebraic Programming*, 79, 6, s. 291 – 294, 2010. ISSN 1567-8326. doi: <http://dx.doi.org/10.1016/j.jlap.2010.04.002>. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S1567832610000287>.
- PEARSON, K. The Problem of the Random Walk. *Nature*, 72, 1865, s. 294, 1905.
- PEETA, S., ZILIASKOPOULOS, A. K. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics*, 1, 3, s. 233–265, 2001. ISSN 1572-9427. doi: 10.1023/A:1012827724856. Dostupné z: <http://dx.doi.org/10.1023/A:1012827724856>.
- PÉREZ-JIMÉNEZ, M. J., ROMERO-CAMPERO, F. J. *P Systems, a New Computational Modelling Tool for Systems Biology*, s. 176–197. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. doi: 10.1007/11880646\_8. Dostupné z: [http://dx.doi.org/10.1007/11880646\\_8](http://dx.doi.org/10.1007/11880646_8). ISBN 978-3-540-46236-1.



- PESCINI, D., BESOZZI, D., MAURI, G., ZANDRON, C. Dynamical probabilistic P systems. *International Journal of Foundations of Computer Science*, 17, 1, s. 183–204, February 2006.
- PRIGORINE, I., HERMAN, R. *Kinetic Theory of Vehicular Traffic*. New York, USA : American Elsevier, 1971.
- ROMERO-JIMÉNEZ, A., GUTIÉRREZ-NARANJO, M. A., PÉREZ-JIMÉNEZ, M.-J. The growth of branching structures with P systems. In GUTIÉRREZ-NARANJO, M. A., PAUN, G., RISCOS-NÚÑEZ, A., ROMERO-CAMPERO, F. J. (Ed.) *Fourth Brainstorming Week on Membrane Computing, Sevilla, January 30 - February 3, 2006. Volume II*, s. 253–266. Fénix Editora, 2006.
- RUSKIN, H. J., WANG, R. *Modeling Traffic Flow at an Urban Unsignalized Intersection*, s. 381–390. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002. doi: 10.1007/3-540-46043-8\_38. Dostupné z: [http://dx.doi.org/10.1007/3-540-46043-8\\_38](http://dx.doi.org/10.1007/3-540-46043-8_38). ISBN 978-3-540-46043-5.
- SHERALI, H. D., PARK, T. Estimation of dynamic origin–destination trip tables for a general network. *Transportation Research Part B: Methodological*, 35, 3, s. 217 – 235, 2001. ISSN 0191-2615. doi: [http://dx.doi.org/10.1016/S0191-2615\(99\)00048-X](http://dx.doi.org/10.1016/S0191-2615(99)00048-X). Dostupné z: <http://www.sciencedirect.com/science/article/pii/S019126159900048X>.
- STEELE, M. J. *Stochastic Calculus and Financial Applications*. Springer-Verlag New York, 2001.
- STEPHANOPOULOS, G., MICHALOPOULOS, P. G., STEPHANOPOULOS, G. Modelling and analysis of traffic queue dynamics at signalized intersections. *Transportation Research Part A: General*, 13, 5, s. 295 – 307, 1979. ISSN 0191-2607. doi: [http://dx.doi.org/10.1016/0191-2607\(79\)90028-1](http://dx.doi.org/10.1016/0191-2607(79)90028-1). Dostupné z: <http://www.sciencedirect.com/science/article/pii/0191260779900281>.
- SUZUKI, Y., FUJIWARA, Y., TAKABAYASHI, J., TANAKA, H. Artificial life applications of a class of p systems: Abstract rewriting systems on multisets. In CALUDE, C., PAUN, G., ROZENBERG, G., SALOMAA, A. (Ed.) *WMP, 2235 / Lecture Notes in Computer Science*, s. 299–346. Springer, 2000. Dostupné z: <http://dblp.uni-trier.de/db/conf/wmp/wmp2000.html#SuzukiFTT00>. ISBN 3-540-43063-6.

- SUZUKI, Y., FUJIWARA, Y., TANAKA, H., TAKABAYASHI, J. Artificial life applications of a class of P systems: Abstract rewriting systems on multisets. In CALUDE, C., PĂUN, G., ROZENBERG, G., SALOMAA, A. (Ed.) *Multiset Processing: Mathematical, Computer Science, and Molecular Computing Points of View*, 2235 / *Lecture Notes in Computer Science*, s. 299–346, Berlin, 2001. Springer-Verlag.
- TOBLER, W. R. A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46, s. 234–240, 1970. ISSN 00130095, 19448287. Dostupné z: <http://www.jstor.org/stable/143141>.
- TSANG, C., HO, T. Passenger flow and station facilities modelling for metro station layout design. In *4th International Conference on Traffic and Transportation Studies*, 2004.
- URBANIK, T., TANAKA, A., LOZNER, B., LINDSTROM, E., LEE, K., QUAYLE, S., BEAIRD, S., TSOI, S., RYUS, P., GETTMAN, D. et al. Signal timing manual. *National Cooperative Research Program*, s. 03–103, 2014.
- VERLAN, S., BERNARDINI, F., GHEORGHE, M., MARGENSTERN, M. Generalized communicating p systems. *Theoretical Computer Science*, 404, 1, s. 170 – 184, 2008. ISSN 0304-3975. doi: <http://dx.doi.org/10.1016/j.tcs.2008.04.008>. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0304397508002582>.
- WORTMAN, R. H., MATTHIAS, J. S. *An Evaluation of Driver Behavior at Signalized Intersections*. Arizona Department of Transportation, 1983.

# PŘÍLOHY

## Práce se softwarovým nástrojem

Tato kapitola ukazuje práci se softwarovým nástrojem (dále jen *simulátor*), který byl vytvořen pro simulaci *transportních P systémů se signalizací, kapacitami a událostmi*. Veškerý kód je přílohou této práce a je volně stažitelný ze stránek [https://github.com/janoskaz/trans\\_p\\_systems](https://github.com/janoskaz/trans_p_systems).

Všechn kód je napsán v jazyce **R**, prostředí pro práci v tomto programovacím jazyce lze stáhnout ze stránek <https://www.r-project.org>. Pro práci se simulátorem stačí základní verze programu ve verzi **3.0.0** a vyšší.

Následující text předpokládá, že uživatel si stáhnul všechny soubory ze stránek [https://github.com/janoskaz/trans\\_p\\_systems](https://github.com/janoskaz/trans_p_systems) a v nezměněné adresářové struktuře je uložil na svůj disk.

Nejprve je třeba nastavit pracovní adresář a načíst zdrojové soubory.

```
# nastavit pracovni adresar
setwd("/path/to/trans_p_systems")

# nacist zdrojove soubory
# soubor signaledTranspPSystems.R obsahuje kod pro nacteni zdrojovych souboru
source("signaledTranspPSystems.R")
```

Všechn kód obsahující definice používaných tříd a funkcí je obsažen v souborech `src/signaledTranspPSyst_classes.R` a `src/signaledTranspPSyst_functions.R`; alternativně je možné nahrát pouze kód těchto dvou souborů:

```
# nacist zdrojove soubory
source('/path/to/trans_p_systems/src/signaledTranspPSyst_classes.R')
source('/path/to/trans_p_systems/src/signaledTranspPSyst_functions.R')
```

Soubor `src/signaledTranspPSyst_classes.R` obsahuje definice tříd pro simulaci P systémů. Základním objektem jsou multimnožiny objektů, uložené v objektech třídy `multiset`. K vytvoření objektu slouží funkce `createMultiset(object, target)`, kde `object` je řetězec reprezentující jeden objekt a `target` je numerický popisník membrány, do níž objekt směřuje. Parametr `target` je nepovinný.

Objekt třídy `multiset` slouží k reprezentaci objektů včetně katalyzátorů.

```

# vytvoreni objektu - vozidlo smerujici do membrany 4
veh = createMultiset("vehicle",4)

# zobrazeni objektu
veh
#An object of class "multiset"
#Slot "obj":
#[1] "vehicle"

#Slot "tar":
#[1] 4

# user friendly vypsani objektu
printMultiset(veh)
#[1] "vehicle(in_4)"

# vytvoreni katalyzatoru
catalyst = createMultiset('c')

```

Objekt třídy `synapse` slouží k reprezentaci synapsí a je vytvořen pomocí funkce `createSynapse(from,to,status)`, kde `from`,`to` jsou numerické popisky membrán, mezi nimiž synapse existuje a nepovinný parametr `status` je logická proměnná, určující stav synapse.

```

# vytvoreni synapse - otevrena synapse mezi membranami 4 a 8
syn = createSynapse(4,8,TRUE)

# zobrazeni objektu
syn
#An object of class "synapse"
#Slot "f":
#[1] 4

#Slot "t":
#[1] 8

#Slot "s":
#[1] TRUE

```

K vytvoření pravidel je nutné nejprve vytvořit objekty, reprezentující pravou a levou stranu pravidla - objekty tříd `ruleRightSide`,`ruleLeftSide`. Objekt třídy `ruleRightSide` obsahuje pouze dvě vlastnosti: `object` značící nově vytvořený multiset a `label` značící popisek membrány, v níž má být objekt vytvořen.

Objekt třídy `ruleLeftSide` obsahuje vlastnosti `object` značící spotřebovaný objekt, `label` značící popisek membrány, z níž je objekt odstraněn, `synapse` značící synapsi, která je k vykonání pravidla nutná a `catalyst` reprezentující katalyzátor reakce. `synapse` a `catalyst` jsou nepovinné parametry.

```

# vytvoreni leve a prave strany pravidla - presunuti objektu veh z membrany 4 do
  membrany 8 pokud existuje otevrena synapse a je pritomny katalyzator

# ruleLeftSide
veh1 = createMultiset("vehicle",8) # objekt smerujici do membrany 8
c = createMultiset("c") # katalyzator
syn = createSynapse(4,8,TRUE)

leftSide = createRLS(label=4,object=veh1,synapse=syn,catalyst=c)

# ruleRightSide
veh2 = createMultiset("vehicle")

rightSide = createRRS(label=8,object=veh2)

```

Objekt třídy `rule` obsahuje kromě objektů pro pravou a levou stranu pravidla rovněž parametry popisující čas, pravděpodobnost výběru pravidla, status synapse před a po použití pravidla.

```

# definice tridy rule
setClass('rule',
  slots=c(
    left='ruleLeftSide', # leva strana
    right='ruleRightSide', # prava strana
    time='numeric', # cas vykonani
    prob='numeric', # pravdepodobnost vyberu
    statusFrom='logical', # stav synapse pred
    statusTo='logical')) # stav synapse po

# definice funkce pro vytvoreni pravidla
createRule = function(lrs,rrs,t=1,p=1,statFrom=logical(0),statTo=logical(0)){
  r <- new('rule',left=lrs,right=rrs,time=t,prob=p,statusFrom=statFrom,statusTo=
    statTo)
  return(r)
}

```

Jediné povinné parametry jsou `left`, `right` – levá a pravá strana pravidla. Pokud není zadán čas výpočtu, je defaultně nastaven na 1, pokud není zadána pravděpodobnost výběru pravidla, je defaultně nastavena na 1. Pokud nejsou zadány stavy synapse před a po použití pravidla, nedochází v průběhu výpočtu ke změně synapse. Pokud jsou stavy synapsí zadány, bude pravidlo chápáno jako pravidlo měnící stav synapsí, i pokud by byly stavy před a po použití pravidla shodné.

Pokud nedochází v průběhu výpočtu ke změně stavu synapse, neměly by parametry `statFrom` a `statTo` být zadány.

```

# vytvoreni pravidla pro premistení vozidla
# cas cesty = 5
# pravdepodobnost vyberu = 1500
# zmena stavu synapse z + na -

```

```

# vytvoreni pravidla
myrule = createRule(
    lrs = leftSide,
    rrs = rightSide,
    t=5,
    p=1500,
    statFrom=TRUE,
    statTo=FALSE)

# print rule (user friendly)
printRule(myrule)
#[1] "[vehicle8|syn48+|c]4 -> [vehicle]8 ; 5, 1500, TRUE, FALSE"
# funkce zahlasí varovani, ktere muze byt ignorovano

```

Function `printRule(rule)` vypíše do konzole zkrácený zápis pravidla ve formě `[vehicle8|syn48+|c]4 -> [vehicle]8 ; 5, 1500, TRUE, FALSE`, kterou lze číst jako „objekt *vehicle* (*in* 8) je přemístěn z membrány 4 do membrány 8, kde se změní na objekt *vehicle*. K přemístění je nutná existence synapse mezi membránami 4 a 8 v otevřeném stavu (+) a přítomnost katalyzátoru *c*. Aplikace pravidla trvá 5 jednotek, pravděpodobnost výběru je 1500 a během aplikace dojde ke změně stavu synapse (4,8) z *TRUE* (+) na *FALSE* (-)“.

Události jsou reprezentovány pomocí objektů třídy `event` a jsou vytvářeny funkcí `createEvent(r,t)` mající dva parametry: pravidlo `r` a čas `t`.

```

# vytvoreni udalosti - premisteni vozidla v case 11
# jako pravidlo je pouzit objekt r, vytvoreny v predchozi ukazce

# definice funkce pro vytvoreni pravidla
evt = createEvent(
    r = myrule
    t = 11)

```

Visící pravidla jsou reprezentována jako objekty třídy `pendingRule` a lze je vytvořit buď přímo funkcí `createPendingRule` nebo z objektu třídy `rule` za použití funkce `createPendingRuleFromRule`.

```

# definice tridy pendingRule
setClass('pendingRule', slots=c(
    catalyst='multiset',
    membraneFrom='numeric',
    right='ruleRightSide',
    time='numeric',
    syn='synapseOpt',
    statusFrom='logical',
    statusTo='logical'))

```

```

# vytvoreni visiciho pravidla z objektu myrule
myPendingRule = createPendingRuleFromRule(myrule)
str(myPendingRule) # struktura objektu
#Formal class 'pendingRule' [package ".GlobalEnv"] with 7 slots
# ..@ catalyst      :Formal class 'multiset' [package ".GlobalEnv"] with 2 slots
# .. .. ..@ obj: chr "c"
# .. .. ..@ tar: num(0)
# ..@ membraneFrom: num 4
# ..@ right         :Formal class 'ruleRightSide' [package ".GlobalEnv"] with 2 slots
# .. .. ..@ memlabel: num 8
# .. .. ..@ obj      :Formal class 'multiset' [package ".GlobalEnv"] with 2 slots
# .. .. .. .. ..@ obj: chr "vehicle"
# .. .. .. .. ..@ tar: num(0)
# ..@ time          : num 4
# ..@ syn           :Formal class 'synapse' [package ".GlobalEnv"] with 3 slots
# .. .. ..@ f: num 4
# .. .. ..@ t: num 8
# .. .. ..@ s: logi TRUE
# ..@ statusFrom   : logi TRUE
# ..@ statusTo     : logi FALSE

```

Pro reprezentaci seznamu pravidel a seznamu objektů slouží třídy `multisets` a `rules`, jedná se o pomocné třídy.

```

# priklad vytvoreni seznamu objektu
msets = new("multisets",objects=list(obj1,obj2,obj3))
# priklad vytvoreni prazdneho seznamu objektu
msets = new("multisets",objects=list())

# priklad vytvoreni seznamu pravidel
rset = new("rules",rules=list(r1,r2,r3))

```

Pravidla a objekty jsou umístěny v membránách, reprezentovaných objekty třídy `membrane`. K vytvoření membrány slouží funkce `createMembrane(memlabel,objects,rules)`, kde `memlabel` je číselný popis membrány, `objects` je objekt třídy `multisets` a `rules` je objekt třídy `rules`. Objekty `multisets`, `rules` mohou obsahovat prázdné seznamy (membrána neobsahuje žádné objekty ani pravidla).

```

# priklad vytvoreni membraby
memb <- createMembrane(memlabel=1,rules=rset,objects=mset)

```

Konečně třída pro vlastní membránový systém je `membraneStructure` a je vytvořena funkcí `createMembraneStructure(membranes,synapses,pendingRules,pendingSynapses,evts)`, kde všechny parametry jsou objekty třídy `list` (běžný seznam v jazyce **R**). Parametry `pendingSynapses` a `evts` (visící synapse a události) jsou nepovinné.

Vlastní simulace je provedena použitím funkce `simulate(ms)`, jejíž parametry

jsou objekt třídy `membraneStructure` a celé číslo `time`, reprezentující počet kroků, po které má výpočet probídat (tento parametr je nepovinný).

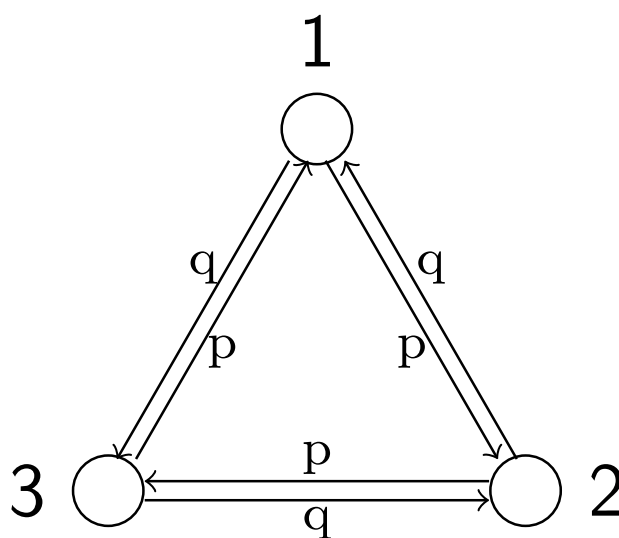
## Náhodná procházka

Použití simulátoru je demonstrováno na příkladu simulace náhodné procházky pomocí *trnaposrtních P systémů*.

Membránový systém je definován tak, jak je uvedeno v kapitole 4.7. Skládá se ze tří membrán, které jsou navzájem spojeny synapsemi. V počátek výpočtu je v membráně 1 umístěn jeden objekt *veh* a každá membrána obsahuje právě dvě pravidla ve tvaru:

$$\left[ veh \mid syn_{(i,j)} \right]_i \xrightarrow{t=1, p=1} \left[ veh \right]_j, \quad (48)$$

Čas transportu mezi dvěma membránami je vždy 1 jednotka a pravděpodobnost transportu do každé ze dvou sousedních membrán je shodný ( $p = q = 1$ ).



Obrázek 23: Membránová struktura pro simulaci jednoduché náhodné procházky

Kód pro definici modelu v simulátoru je:

```
# vytvorit objekt
vehicle = createMultiset('veh')

# membrana 1
# vytvorit synapse
syn12 <- createSynapse(1,2)
syn13 <- createSynapse(1,3)
# leva strana pravidel
lrs12 <- createRLS(label=1, object=vehicle, synapse=syn12)
lrs13 <- createRLS(label=1, object=vehicle, synapse=syn13)
```



```

# prava strana pravidel - neobsahuje synapsi
rrs12 <- createRRS(label=2, object=vehicle)
rrs13 <- createRRS(label=3, object=vehicle)
# kompletne pravidla
r1a <- createRule(lrs=lrs12, rrs=rrs12, t=1, p=1) # pravdepodobnost p v jednom smeru
r1b <- createRule(lrs=lrs13, rrs=rrs13, t=1, p=1) # pravdepodobnost q ve druhem smeru
# membrana vctne obsahu - obsahuje objekt vehicle
m1 <- createMembrane(memlabel=1,
                      objects=new('multisets',objects=list(vehicle)),
                      rules=new('rules',rules=list(r1a,r1b)))

# membrana 2
# synapse
syn23 <- createSynapse(2,3)
syn21 <- createSynapse(2,1)
# leva strana pravidel
lrs23 <- createRLS(label=2, object=vehicle, synapse=syn23)
lrs21 <- createRLS(label=2, object=vehicle, synapse=syn21)
# prava strana pravidel
rrs23 <- createRRS(label=3, object=vehicle)
rrs21 <- createRRS(label=1, object=vehicle)
# kompletne pravidla
r2a <- createRule(lrs=lrs23, rrs=rrs23, t=1, p=1) # pravdepodobnost p v jednom smeru
r2b <- createRule(lrs=lrs21, rrs=rrs21, t=1, p=1) # pravdepodobnost q ve druhem smeru
# membrana vctne obsahu - neobsahuje objekty, jen pravidla
m2 <- createMembrane(memlabel=2,
                      objects=new('multisets',objects=list()),
                      rules=new('rules',rules=list(r2a,r2b)))

# membrana 3
# synapse
syn31 <- createSynapse(3,1)
syn32 <- createSynapse(3,2)
# leva strana pravidel
lrs31 <- createRLS(label=3, object=vehicle, synapse=syn31)
lrs32 <- createRLS(label=3, object=vehicle, synapse=syn32)
# prava strana pravidel
rrs31 <- createRRS(label=1, object=vehicle)
rrs32 <- createRRS(label=2, object=vehicle)
# kompletne pravidla
r3a <- createRule(lrs=lrs31, rrs=rrs31, t=1, p=1) # pravdepodobnost p v jednom smeru
r3b <- createRule(lrs=lrs32, rrs=rrs32, t=1, p=1) # pravdepodobnost q ve druhem smeru
# membrana vctne obsahu - neobsahuje objekty, pouzije pravidla
m3 <- createMembrane(memlabel=3,
                      objects=new('multisets',objects=list()),
                      rules=new('rules',rules=list(r3a,r3b)))

# seznam vsech synapsi
synapses = list(syn12,syn31,syn23,syn21,syn31,syn32)

# kompletne membranovy system
membraneSystem <- createMembraneStructure(
  membranes=list(m1,m2,m3),
  synapses=synapses,
  evts=list(),
  pendingRules=list())

```

Simulace již definované systému se provede zavoláním funkce `simulate` s volitelným parametrem `time`, označujícím počet kroků, po které má simulace běžet. U tohoto konkrétního systému je nutné parametr uvést, protože jinak k dokončení výpočtu nikdy nedojde.

```
# simulace nahodne prochazky po dobu 20 kroku
sim <- simulate(membraneSystem,20)
```

Výsledkem výpočtu je objekt `sim`, což je seznam konfigurací systému ve všech časových krocích. Každá jednotlivá konfigurace udává seznam obsahů jednotlivých membrán (sestává z popisku membrány a seznamu objektů v membráně).

```
# objekt sim je seznam 20 konfiguraci pro jednotlivé casove kroky
length(sim)
#[1] 20

# struktura objektu reprezentujiciho prvni konfiguraci systemu
str(sim[[1]])
#List of 3
# $ :List of 2
# ..$ : num 1
# ..$ : list()
# $ :List of 2
# ..$ : num 2
# ..$ :List of 1
# .. ..$ :Formal class 'multiset' [package ".GlobalEnv"] with 2 slots
# .. .. .. @ obj: chr "veh"
# .. .. .. @ tar: num(0)
# $ :List of 2
# ..$ : num 3
# ..$ : list()
```

Z objektu `sim` lze získat polohu objektu `veh` v jednotlivých časových krocích (pomocí standardních funkcí programu **R**)

```
# vysledky simulace jako matice
M <- t(simplify2array(lapply(sim,function(y){
  unlist(lapply(y,function(x){return(length(x[[2]))}))
})))

# ukazka matice
# kazdy sloupec reprezentuje jednu membranu
# 1 = objekt pritomny, 0 = neni pritomny
head(M)
#      [,1] [,2] [,3]
#[1,]  0   1   0
#[2,]  1   0   0
#[3,]  0   1   0
#[4,]  1   0   0
#[5,]  0   1   0
#[6,]  0   0   1
```

```

# vypocist cestu grafem
currentPosition <- 1 # ve ktorej membrane
currentNumber <- 0 # hodnota
res <- c(0,0) # pro ulozeni cesty, v kroce 0 hodnota 0

# pro kazdy casovy krok
for (i in 1:nrow(M)){
  # ve ktorej membrane se objekt nachazi
  w <- which(M[i,] == 1)
  # v membrane 1
  if (w == 1){
    # predchozi membrana 2 - krok -1
    if (currentPosition == 2){
      currentNumber <- currentNumber-1
    # predchozi membrana 3 - krok +1
    } else {
      currentNumber <- currentNumber+1
    }
  # v membrane 2
  } else if (w == 2){
    # predchozi membrana 3 - krok -1
    if (currentPosition == 3){
      currentNumber <- currentNumber-1
    # predchozi membrana 1 - krok +1
    } else {
      currentNumber <- currentNumber+1
    }
  # v membrane 3
  } else {
    # predchozi membrana 1 - krok -1
    if (currentPosition == 1){
      currentNumber <- currentNumber-1
    # predchozi membrana 2 - krok +1
    } else {
      currentNumber <- currentNumber+1
    }
  }
  currentPosition <- w
  res <- rbind(res,c(i,currentNumber))
}

# ukazka cesty grafem
head(res)
#      [,1] [,2]
#res    0    0
#      1    1
#      2    0
#      3    1
#      4    0
#      5    1

# vykreslit do grafu
plot(res,type="l",xlab="krok",ylab="hodnota")

```

Další, pokročilejší simulace jsou dostupné na stránce [https://github.com/janoskaz/trans\\_p\\_systems/tree/master/examples](https://github.com/janoskaz/trans_p_systems/tree/master/examples).



**KATEDRA GEOINFORMATIKY**

Univerzita Palackého v Olomouci | Přírodovědecká fakulta

# **VYUŽITÍ P SYSTÉMŮ PRO MODELOVÁNÍ PROSTOROVÝCH JEVŮ**

## **AUTOREFERÁT DISERTAČNÍ PRÁCE**

Studijní program: P1314 Geografie

Obor studia: 1302V011 Geoinformatika a kartografie

Školitel: Doc. Mgr. Jiří DVORSKÝ, Ph.D.

**Mgr. Zbyněk Janoška**

## **APPLICATIONS OF P SYSTEMS IN SPATIAL MODELLING**

### **Ph.D. THESIS SUMMARY**

Study Programme: Geography

Specialization: Geoinformatics and Cartography

Supervisor: Doc. Mgr. Jiří DVORSKÝ, Ph.D.

### **Department of Geoinformatics**

Faculty of Science, Palacký University Olomouc

**Olomouc 2017**

*Disertační práce byla vypracována v distanční formě doktorského studia na Katedře geoinformatiky Přírodovědecké fakulty Univerzity Palackého v Olomouci.*

*Dissertation thesis was compiled within Ph.D. study at the Department of Geoinformatics, Faculty of Science, Palacký University Olomouc.*

**Předkladatel / Submitter:**

Mgr. Zbyněk Janoška

**Školitel / Supervisor:**

Doc. Mgr. Jiří DVORSKÝ, Ph.D.

Katedra geoinformatiky

Přírodovědecká fakulta Univerzity Palackého v Olomouci

17. listopadu 50

771 46 Olomouc

**Oponenti / Opponents:**

doc. RNDr. Petr Kubíček, CSc. (Masarykova univerzita, Brno)

doc. Ing. Petr Sosík, Dr. (Slezská univerzita v Opavě)

doc. Ing. Miloslav Hub, Ph.D. (Univerzita Pardubice)

Autoreferát byl zaslán dne / Summary was posted on: 25.4.2017

Obhajoba disertační práce se koná dne [redacted] před komisí

pro obhajoby disertačních prací doktorského studia v oboru P1314 Geografie, studijním oboru 1302V011 Geoinformatika a kartografie, v prostorách Katedry geoinformatiky Přírodovědecké fakulty Univerzity Palackého v Olomouci, 17. listopadu 50, 771 46 Olomouc.

The defence of the dissertation thesis will be held on [redacted] at the commission for the defence of dissertation thesis of Ph.D. degree in study programme P1314 Geography, specialization Geoinformatics and cartography, in the premises of the Department of Geoinformatics, Faculty of Science, Palacký University Olomouc, 17. listopadu 50, 771 46 Olomouc.

*S disertační prací je možno se seznámit na studijním oddělení Přírodovědecké fakulty Univerzity Palackého v Olomouci, 17. listopadu 12, 77 46 Olomouc.*

*The dissertation thesis is available at the Study Department, Faculty of Science, Palacký University in Olomouc, 17. listopadu 12, 771 46 Olomouc.*

© Zbyněk Janoška, 2017

ISSN [redacted]

## Obsah

1.	Abstrakt	4
2.	Cíle práce řešerše	5
3.	Model transportu vozidel v síti	10
4.	Simulace	12
5.	Transportní P systém s vozidly	28
6.	Výsledky	31
7.	Diskuze	36
8.	Závěr	38
9.	Použité zdroje	43
	Odborný životopis autora	43
	Seznam vybraných publikací autora	44

## Contents

	Curriculum vitae	43
	Author's selected publications	44
	Annotation	45
	Summary	46

## 1. Abstrakt

P systémy představují výpočetní model, inspirovaný strukturou a fungováním živé buňky. Disertační práce se zabývá možností využití P systémů pro modelování transportu. Téma transportu v membránových systémech bylo široce zkoumáno, téměř výhradně však z pohledu biochemického. Transport v geografickém prostoru je odlišný zejména proto, že se neděje v hierarchické struktuře oddělení (kterou klasické P systémy představují), ale v rovině (případně ve vyšších rozměrech). Práce navazuje na řadu publikací, které se transportem v P systémech zabývaly a představuje dva modely transportu, jež byly sestaveny na základě inspirace fungováním dopravy osob a zboží. Oba modely byly sestaveny za účelem provádění simulací a nikoli výpočtů.

*Transportní P systém se signalizací, kapacitami a událostmi* modeluje transport objektů v síti, jejíž uzly tvoří membrány. Inovativním rozšířením nad rámec dříve publikovaných variant P systémů jsou události, které mají podobu pravidel, která jsou použita v daný časový krok. Pravidla slouží k reprezentaci jednorázových událostí, jakými jsou například přerušení komunikace nebo náhlé zvýšení počtu objektů v části systému. Využití modelu je demonstrováno na několika příkladech, včetně zcela teoretických úloh (náhodná procházka v grafu) i reálných systémů (model dopravy na křižovatce, transport říční sítí). Je dokázána turingovská úplnost navrženého modelu a je vytvořen simulátor v jazyce R.

*Transportní P systém s nosiči* modeluje transport objektů pomocí vozidel, přičemž inovativním prvkem je použití mobilních membrán reprezentujících vozidla.

Klíčová slova:

P systémy, transport, doprava



## 2. Cíle práce a řešerše

Hlavním cílem práce je sestavit model dopravy v síti založený na formalismu P systémů. Ačkoli je téma transportu v rámci P systémů intenzivně zkoumáno, nebyly publikovány žádné práce, které by se zabývaly dopravou nebo přepravou cestujících a nákladu. Téma transportu je u P systémů zkoumáno buď na úrovni teoretických modelů nebo jsou modelovány biochemické jevy, které se od dopravy v mnohém liší; zejména tím, že transport probíhá v hierarchicky uspořádaných systémech (buňkách) nebo v tkáních, kde nehraje roli vzdálenost a závisí na sousedství.

Vzhledem k relativní novosti tématu membránových výpočtů je součástí práce podrobnější popis této výpočetní metody. Rešerše literatury je rozšířena nad rámec popisu základního modelu a odkazů na literaturu a obsahuje popis variant P systémů, které jsou relevantní pro danou problematiku a zabývají se transportem nebo reprezentací prostoru a prostorových vazeb. Jelikož škála reálných dopravních systémů jimiž se lze zabývat je široká, zaměřuji se v této práci na dva modely, shrnuté v dílčích cílech:

**DC1** - model transportu vozidel v síti

**DC2** - model transportu cestujících v systémech veřejné dopravy

**Model transportu vozidel v síti** je typickou úlohou modelování dopravy. Model pohybu vozidel v síti obsahující křižovatky se signalizací se navržen ve dvou krocích. Nejprve je představen jednoduchý model transportu v síti se signalizovanými křižovatkami a poté rozšiřující model, který umožňuje pracovat s kapacitami křižovatek a jednorázovými událostmi. Model vychází zejména z generalizovaných komunikačních P systémů (Verlan et al., 2008) a tkáňovitých P systémů se stavy komunikačních kanálů (Freund et al., 2004), které rozšiřuje tak, aby bylo možné realisticky reprezentovat transport vozidel v síti. Kapacity křižovatek umožňují regulovat počet přepravovaných objektů které jsou ve výpočetním kroku „obslouženy“, zatímco události umožňují simulovat procesy, které jsou jednorázové a vedou k často permanentním změnám systému (přerušeni komunikace, náhlé zvýšení počtu přepravovaných objektů apod.).

Je dokázáno, že navržený model je turingovský úplný.

Součástí práce je softwarový nástroj, který uživateli umožňuje navrhnout libovolný P systém (dle popsaného modelu) a provést v něm výpočet. Software je řešen jako sada skriptů v jazyce R. Příloha práce obsahuje návod, jak se skripty pracovat a jak interpretovat výsledky.

Je provedeno několik simulací modelových systémů, které demonstrují využitelnost pro modelování transportu. Simulace ukazují použití navrženého modelu jak pro řešení zcela teoretických úloh, mezi něž patří například náhodná procházka a z ní odvozená hra ruinování hráče, tak i simulace transportu s reálných sítích.

**Transportu cestujících v systémech veřejné dopravy** představuje jednodušší dopravní systém, než doprava individuální. Jsou pevně dány body, jimiž cestující vstupují a vystupují ze systému (zastávky, stanice) a k transportu dochází pomocí velkokapacitních prostředků, bez nichž není pohyb v síti možný. Nejjednodušší typ těchto systémů představují lanové a podzemní dráhy. Vzhledem k tomu, že objekty v P systémech nefungují jako agenti a nedochází k rozhodování, je možné tento model abstrahovat pro simulaci pohybu zboží, materiálu, informací apod. Formální model transportu cestujících v sítích veřejné dopravy navazuje na varianty P systémů, inspirované transportem látek v buňkách a buněčných strukturách (Päun a Päun, 2002; Verlan et al., 2008) a rozšiřuje je o další prvky, umožňující práci s pravděpodobností (Suzuki et al., 2000), časem (Cavaliere a Zandron, 2006) a pohybem membrán v systému (Krishna a Päun, 2005).

Kromě základního modelu jsou diskutována rozšíření, nezbytná pro věrohodnou reprezentaci přepravy cestujících. Mezi tyto rozšíření patří nutnost reprezentovat trasy vozidel, trasy cestujících, časovou variabilitu v dopravní poptávce a jízdní řády.

Jelikož membránové systémy nejsou obecně známým výpočetním modelem, nejprve jsou v rešerši uvedeny základní informace o tomto výpočetním modelu včetně rozšíření nutných pro realistickou reprezentaci pohybu v P systémech.

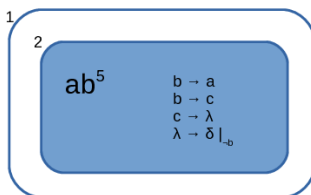
Membránové výpočty jsou oblastí počítačové vědy, která se snaží abstrahovat výpočetní principy z chování a struktury živých buněk, z jejich uspořádání a organizace do tkání a struktur vyššího řádu. P systémy poprvé představil (Päun, 1998) jako distribuovaný paralelní výpočetní model inspirovaný funkcí živé buňky. P systémy pracují s multimnožinami objektů, které jsou v systému lokalizovány. Objekty a pravidla, která s nimi manipulují, jsou zapouzdřeny v membránách. Ústřední roli hraje komunikace mezi jednotlivými membránami (a okolím systému).

Nejvýraznějším znakem P systémů je jejich organizace do oddělených částí.

Dalším pojmem, který byl zmíněn, je multimnožina objektů. V živé buňce obsahují jednotlivá oddělení buňky substance (ionty, menší molekuly, makromolekuly) plovoucí v kapalném roztoku. Nelze zde

najít uspořádání - vše je blízké všemu, záleží na koncentraci a množství kopií jednotlivých substancí. Abstrahujeme-li tyto principy matematicky, dostaneme množiny objektů a jejich počty - tedy multimnožiny. Biochemické reakce probíhající v odděleních buňky jsou reprezentovány pomocí pravidel s ohledem na reprezentaci multimnožiny. Mají formu  $u \rightarrow v$ , kde  $u$  a  $v$  reprezentují multimnožiny objektů. Příkladem pravidla může být  $aab \rightarrow abcc$ . Při výpočtu v P systému se uplatňují následující principy:

- nedeterminismus: obvykle je v membráně obsaženo velké množství objektů a s nimi spřaženo mnoho pravidel. Zásadou je, že pravidla jsou vybírána a na objekty aplikována nedeterministicky, tedy v náhodném pořadí.
- paralelismus: biochemické reakce se dějí nejen nedeterministicky ale také paralelně. Pokud dvě látky mohou spolu reagovat, tak reagují nejen dvě jejich molekuly, ale všechny dostupné molekuly. V P systémech jsou v každém kroku aplikována všechna pravidla, která mohou být aplikována, na všechny dostupné objekty ve všech membránách.



Obr. 1 Ukázka P systému pro simulaci hodu kostkou

## Formální definice

Systémy založené na evolučních pravidlech (přepisujících objekty v multimnožinách) popsaných výše jsou nazývány *transition P systémy*. Formálně, *transition P systém* je konstrukt ve tvaru (Păun, 1998):

$$\Pi = (O, C, \mu, w_1, w_2, \dots, w_m, R_1, R_2, \dots, R_m, i_o)$$

kde:

- $O$  je konečná a neprázdná abeceda objektů,
- $C \subset O$  je množina katalyzátorů,
- $\mu$  je membránová struktura, skládající se z  $m$  membrán, označených  $1, 2, \dots, m$ ; říkáme, že membránová struktura (a tedy systém) je stupně  $m$ ,
- $w_1, w_2, \dots, w_m$  jsou řetězce z  $O$  reprezentující objekty přítomné v regionech  $1, 2, \dots, m$  membránové struktury,

- $R_1, R_2, \dots, R_m$  jsou pravidla asociovaná s regiony  $1, 2, \dots, m$  membránové struktury,
- $i_o$  je buď jeden z regionů  $1, 2, \dots, m$  a pak je tento region výstupním regionem systému nebo je  $0$  a pak je výsledek výpočtu uložen ve vnějším prostředí systému.

Pravidla jsou ve tvaru  $u \rightarrow v$  nebo  $u \rightarrow v\delta$ , s  $u \in O$  a  $v \in (O \times Tar)$ , kde  $Tar = \{here, in, out\}$ . Katalytická pravidla mají tvar  $uc \rightarrow vc$ , kde  $c$  má roli katalyzátoru reakce - sám se nemění ani nespotebovává, ale umožňuje změnu jiných objektů.

## Rozšíření P systémů

Základním znakem P systémů jsou paralelismus a nedeterminismus. Všechna pravidla, která mohou být aplikována jsou aplikována, ve zcela náhodném pořadí. Nedeterminismus se projevuje jak při výběru objektů tak i membrán, pokud například dochází k přemístění objektů v rámci membránové struktury. Omezení jak nedeterminismu, tak paralelismu je atraktivní zejména s ohledem pro simulaci reálných systémů, kde předpoklad absolutního nedeterminismu je často nerealistický. Základními způsoby omezení nedeterminismu jsou *určení priority pravidel*, *kontrola prostupnosti membrány* (Madhu a Krithivasan, 2001), *polarizace membrány* (Alhazov et al., 2004), *definice cílové membrány* a řada způsobů práce s *pravděpodobností* (Obtulowicz a Păun, 2003; Suzuki et al., 2001).

Základní organizace membrán v P systémech je hierarchická. Přirozenou modifikací je umístění membrán do sítě. Toto uspořádání imituje organizaci buněk to tkání a modely se nazývají tkáňovité P systémy (Martín-Vide et al., 2002a). V základní formě je několik (elementárních) membrán rozmístěno v prostředí a komunikují spolu za pomoci symportních/antiportních pravidel (Păun a Păun, 2002).

## Reprezentace prostorových vazeb v P systémech

Při práci s prostorem v P systémech se objevují dva přístupy. První z nich je založený na tom, že jediným lokalizovaným prvkem modelu je membrána. Tento přístup lze nazvat jako **síťový**. Objekty a pravidla nejsou lokalizovány, membrány však reprezentují objekty reálného světa a jsou tudíž prostorově lokalizované. Prostorové informace jsou pouze atributem membrán. Každá membrána pak reprezentuje prostorově lokalizovaný objekt. Takovýto model již dokáže popsat sousedství membrán  $i, j$ , které je

vyjádřeno existencí synapse  $(i; j)$  ale nepracuje se vzdáleností. Tu je možné zahrnout při použití vah, přiřazených každé synapsi. Namísto synapse  $(i; j)$  je použita  $(i; j; w_{ij})$ , kde  $w \in R$  vyjadřuje cenu použití synapse (Besozzi et al., 2007).

Druhý přístup reprezentují **prostorové P systémy** (Barbuti et al., 2011). Membrány jsou umístěny ve dvojrozměrném prostoru. Membrány mají obdélníkový tvar a mohou být do sebe vnořeny. Vždy existuje vnější membrána, v níž jsou všechny membrány obsaženy. Každý objekt v membráně má svou pozici, což znamená, že v jedné membráně se může vyskytovat více objektů stejného označení na různých pozicích. Pravidla, která objekty zpracovávají obsahují i indikátor změny pozice.

### 3. Model transportu vozidel v síti

Předkládaný model má za cíl modelovat chování individuálních vozidel na křižovatkách v síti, chování vozidel na přímých úsecích je ignorováno. Membrány jsou uspořádány do orientovaného grafu, podobně jako v generalizovaných komunikačních P systémech (Verlan et al., 2008). Synapse mezi membránami slouží jako komunikační kanály. Přímý transport mezi membránami, které nejsou spojeny synapsí, není možný. Synapse jsou ohodnoceny váhou, která se rovná času, nutnému k přesunu mezi membránami. Synapse jsou jednosměrné a mezi dvěma membránami může existovat maximálně jedna synapse v každém směru.

V dopravní síti mohou být křižovatky vybaveny (světelnou) signalizací, která určuje, které cesty je možné použít. Signalizaci lze v P systémech reprezentovat buď jako polarizaci membrán a umožnit transport jen objektů se shodnou polarizací (tento způsob jsme použili v dřívější práci, viz (Dvorský et al., 2012)), nebo použít princip stavů komunikačních kanálů (Freund et al., 2004), kde jsou synapsím spojujícím membrány přiřazeny stavy a objekty jimi mohou procházet pouze pokud jsou synapse v určeném stavu. Tento typ reprezentace je vhodnější, protože umožňuje aby z jedné membrány vedlo více otevřených komunikačních kanálů současně a je zde použit. Předkládaný model je zjednodušený v tom, že rozeznává pouze dva stavy komunikačních kanálů - otevřený a uzavřený. Oproti tkáňovitým P systémům se stavy komunikačních kanálů (Freund et al., 2004) nejsou používána symportní a antiportní pravidla, nýbrž pravidla uniportní (Verlan et al., 2008). Je přesouván vždy jeden objekt, jehož kvalita se během transportu může měnit (během transportu se objekt  $a$  může změnit v  $b$ ).

Vozidla jsou v navrženém modelu reprezentována jako objekty. Je možné modelovat různé druhy vozidel (osobní automobily, nákladní automobily, autobusy, jednostopá vozidla apod.). U objektů může být přítomný indikátor cíle, nesoucí informaci o následující membráně, do níž objekt směřuje.

#### Formální definice

*Transportní P systém* je definován jako konstrukt:

$\Pi = (O, C, \mu, w_1, \dots, w_m, K, syn, syn_p, (s_{(i,j)})_{(i,j) \in syn}, R, R_p, Evt)$ , kde

- $O$  je abeceda objektů. Objekt  $u \in O$  může obsahovat indikátor cíle -  $u(in_j) | j \in 1, \dots, m$ , kde  $j$  je popiskem membrány,

- $C \subset O$  je množina katalyzátorů,
- $\mu$  je membránová struktura, sestávající z  $m$  membrán, opatřených popisky  $1, \dots, m$ . Membrány jsou elementární a nachází se všechny přímo ve vnějším prostředí,
- $w_1, \dots, w_m$  jsou multimnožiny objektů, nacházejících se na počátku výpočtu v jednotlivých membránách  $1, \dots, m$ ,
- $K = \{s^+; s^-\}$  je množina stavů synapsí. Stav ( $s^+$ ) reprezentuje synapsi, která je otevřená, zatímco ( $s^-$ ) reprezentuje synapsi zavřenou,
- $syn \subseteq \{(i, j, t, p) | i, j \in \{1, \dots, m\}, \mathcal{X} = j, t, p \in N^+ \}$  je množina synapsí mezi membránami. Každá synapse obsahuje informaci o času  $t$  a pravděpodobnosti  $p$ ,
- $syn_p$  je množina visících (pending) synapsí,
- $s_{(i,j)}$  jsou výchozí stavy synapsí  $(i, j) \in syn$ ,
- $R = R_1, \dots, R_m$  je množina pravidel asociovaných s jednotlivými membránami systému,
- $R_p$  je množina visících (pending) pravidel,
- $Evt$  je množina událostí, přičemž událost je dvojice  $(R_{evt}, t)$ ,

Tvar pravidel z množiny  $R$  je

$$\left[ u, k \text{ syn}'_i \right]_t, p, s' \rightarrow s'' \left[ v \right]_j$$

a událost z množiny  $Evt$  je dvojice  $(R_{Evt}, t)$ , kde  $R_{Evt}$  je pravidlem ve výše uvedené zápisu a  $t$  je celé kladné číslo. Každé pravidlo může obsahovat maximálně jeden objekt  $u \in O$  a jeden katalyzátor  $k \in C$  na levé straně pravidla a maximálně jeden objekt  $v \in O$  na pravé straně pravidla. Každá událost je vykonána v průběhu výpočtu maximálně jedenkrát a to v časovém kroku  $t$ .

## Softwarový nástroj

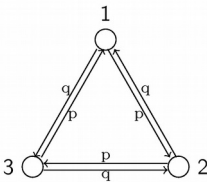
Pro účely simulací v navrženém modelu transportních P systémů se signalizací byl vytvořen softwarový nástroj v programovacím jazyku R. Jedná se o sadu skriptů, stažitelných ze stránek [https://github.com/janoskaz/trans\\_p\\_systems](https://github.com/janoskaz/trans_p_systems).

Nástroj umožňuje konfigurovat libovolný membránový systém, který odpovídá modelu představenému v této práci a provést v něm výpočet.

## 4. Simulace

### Jednoduchá náhodná procházka

Náhodná procházka je jednoduchý stochastický proces, který formalizuje intuitivní proces provádění náhodných kroků (Pearson, 1905; Feller, 1968) v určitém matematickém prostoru, například množině celých čísel. Jednoduchá náhodná procházka je náhodná procházka v jednom rozměru, tedy na přímce. Teorii náhodných procházek lze aplikovat v oborech finančnictví (Steele, 2001), biologie (Codling et al., 2008) nebo fyziky (zejména kinetiky kapalin a plynů) (Blumen et al., 1986) a přes svou jednoduchost představuje matematický model s rozsáhlým aplikačním potenciálem.



Obr 2. Membránový systém pro simulaci jednoduché náhodné procházky

Transportní P systémy mohou sloužit jako jednoduchý simulátor náhodných procházek. Obrázek 2 zobrazuje navrženou membránovou strukturu. Skládá se ze tří membrán (1,2,3), které jsou mezi sebou spojeny pomocí synapsí. Čas cesty po synapsi je vždy roven jedné. Každá synapse má přiřazenu hodnotu pravděpodobnosti použití synapse ( $p$ ,  $q$ ). Obecně se mohou hodnoty  $p$  a  $q$  lišit, pro náhodnou procházku však platí  $p = q = 0,5$ .

Všechny membrány obsahují pravidla

$$\left[ veh, syn_{(i,j)} \right]_i t=1, p \left[ veh \right]_j ,$$

kteřá popisují přemístění objektu  $veh$  z membrány  $i$  do membrány  $j$ . Podmínkou je existence synapse  $(i, j)$ , která obě membrány spojuje.



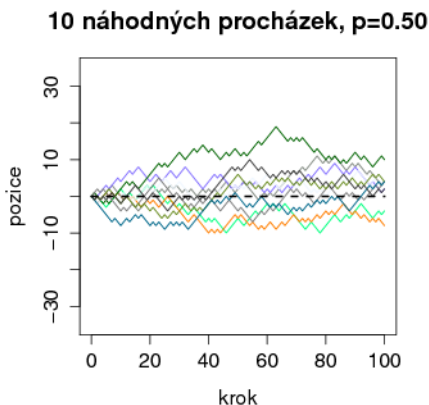
Žádná další pravidla nejsou v žádné z membrán přítomna. Každá membrána obsahuje 2 pravidla, popisující transport objektu *veh* z membrány jedním nebo druhým směrem.

Na počátku výpočtu je jeden objekt *veh* přítomen v membráně 1, ostatní membrány jsou prázdné.

Během výpočtu dochází k přemísťování objektu *veh* membránovou strukturou, přičemž nedojde k zastavení systému. Proto je nutné stanovit počet  $t$  časových kroků, po které bude systém počítat.

Výsledkem výpočtu je seznam konfigurací v každém kroku simulace. Konfigurace udává polohu objektu *veh* v rámci membránové struktury. Vzhledem k tomu, že systém obsahuje pouze jediný objekt, je možné na základě po sobě jdoucích konfigurací systému vysledovat, která pravidla byla použita a kterými synapsemi byl objekt přemísťován.

Na základě pozice objektu *veh* v po sobě jdoucích krocích je dopočítána cesta objektu systémem. Předpokládáme, že na počátku výpočtu je pozice objektu 0 a každým krokem se posune o jednotku v kladném nebo záporném směru. Výsledky 10 běhů programu po dobu 100 kroků jsou na obrázku 3.



Obr 3. 10 náhodných vycházek

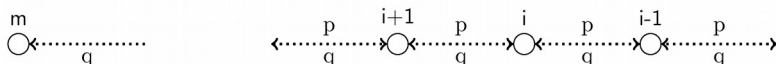
## Ruinování hráče

Ruinování hráče (Gambler's ruin) je statistický koncept, který je nejčastěji vyjádřen jako pravděpodobnost, že hráč s počátečním jměním a konstantním vkladem skončí s nulou za předpokladu, že hráč ukončí hru jen pokud má celý bank, nebo nemá nic.

Předpokládejme, že hrají dva hráči ( $a$  a  $b$ ), jejichž počáteční jmění je  $A$  a  $B$  jednotek, vklad je jedna jednotka a pravděpodobnosti, že hráči vyhrájí jedno kolo hry jsou  $p$  a  $q = 1 - p$ .

Při každém kole hry se hraje o jednu jednotku jmění a to do té doby, než je jeden z hráčů zruinován (a druhý vlastní vše). Za předpokladu, že pravděpodobnosti jsou shodné je modelem tohoto problému jednoduchá náhodná procházka. V rámci transportních P systémů lze ruinování hráče modelovat následovně:

Označme jako  $N = A + B$  celkové jmění obou hráčů. Počet membrán membránového systému je  $m = N + 1$  a popisky membrán jsou  $0, \dots, m$ . Membrány  $i \in \{1, \dots, m-1\}$  jsou synapsí spojeny vždy se dvěma membránami  $i-1, i+1$ . Délky všech synapsí jsou rovny jedné. Pravděpodobnosti asociované se synapsí jsou  $p$  pro synapsi  $(i, i+1)$ , resp.  $q = 1 - p$  pro synapsi  $(i, i-1)$ . Z membrán  $0$  a  $m$  nevedou žádné synapse, nicméně existují synapse  $(1, 0)$  a  $(m-1, m)$ .



Obr 4. P systém pro modelování ruinování hráče

Na počátku výpočtu neobsahuje žádná z membrán objekty s výjimkou té membrány, jejíž popisek  $i$  odpovídá hodnotě  $A$  jmění jednoho z hráčů. Tato membrána obsahuje jeden objekt  $veh$ .

Membrány  $i \in \{1, \dots, m-1\}$  obsahují pravidla ve tvaru:

$$\left[ veh, syn_{(i,j)} \right]_i, t=1, p \left[ veh \right]_j,$$

což je stejné pravidlo, které obsahuje systém pro simulaci náhodné procházky. Membrány  $0, m$  neobsahují žádná pravidla.

Výpočet systému pokračuje do té doby, než je objekt  $veh$  přemístěn do některé z membrán  $0, m$ . Pak je výpočet ukončen, protože nejsou žádná pravidla, která by mohla být v systému aplikována.

Výhodou použití P systému pro simulaci ruinování hráče je paralelismus výpočtního modelu. Do zdrojové membrány může být umístěn libovolný počet  $n$  objektů *veh* a výsledek výpočtu bude odpovídat  $n$  simulacím systému s jediným hráčem. Vzhledem k tomu, že nezáleží na průběhu konkrétní hry (posloupnosti membrán, které jsou objektem navštíveny), ale pouze výsledku (hráč vše vyhrál nebo vše prohrál), lze P systémy pro tento účel dobře použít.

Podíl počtu her, ve kterých vyhrál hráč  $a$  lze porovnat s očekávanými hodnotami (dle (Epstein, 2010)). Počáteční jmění hráče  $a$  je 5, hráče  $b$  je 10.

Tab. 1. Simulace ruinování hráče, 1000 her, počáteční jmění 5 a 10

Pravděpodobnost	0,30	0,40	0,45	0,49	0,50	0,55	0,60
Očekáváno (%)	0,0	1,5	9,0	26,9	33,3	66,7	87,0
Pozorováno (%)	0,0	1,7	9,6	27,2	32,3	70,4	87,7

Hodnoty nasimulované pomocí P systémů odpovídají hodnotám vypočteným. Kromě pravděpodobnosti zruinování, která lze exaktně vypočítat a není nutné ji simulovat, můžou simulace sloužit ke zkoumání například délky trvání hry. Pokud se jedná o férovou hru ( $p = q = 0,5$ ) a tedy vlastně náhodnou procházku, zůstává pravděpodobnost zruinování hráče stejná, pokud se nemění poměr počátečního jmění. Pravděpodobnost zruinování je stejná, pokud je počáteční jmění hráčů 1 a 2, 2 a 4, 5 a 10 apod.

(Steele, 2001) uvádí výpočet očekávané délky trvání, přičemž například při férové hře, kdy oba hráči mají stejnou pravděpodobnost výhry a počáteční jmění hráčů je  $A = 1$ ,  $B = 99$  je očekávané trvání hry 99 kol. Přitom šance, že hráč bude zruinován již po jednom kole, je 50 %.

Porovnání očekávané délky trvání hry a rozdělení délky her u 1000 simulací ukazuje Tabulka 2.

Tab. 2 Délka hry, 1000 her,  $p = q$

A	B	Očekávané trvání	Kvantil simulovaného rozdělení						
			50	75	90	95	97,5	99	100
1	2	2	2	2	4	5	6	7	14
2	4	8	6	10	16	22	28	32	42

3	6	18	13	24	38	49	59	75	145
4	8	32	22	42	68	88	100	122	286
5	10	50	38	72	115	139	168	195	279
1	99	99	3	9	59	311	1092	2773	6751

Prvních 5 řádků tabulky obsahuje výsledky simulací, kdy poměr  $A/B = 1/2$ . Mění se pouze absolutní hodnota parametrů  $A$  a  $B$ . Z výsledků je zřejmé, že charakteristiky rozdělení délky trvání rostou rychleji než hodnoty  $A$  a  $B$ .

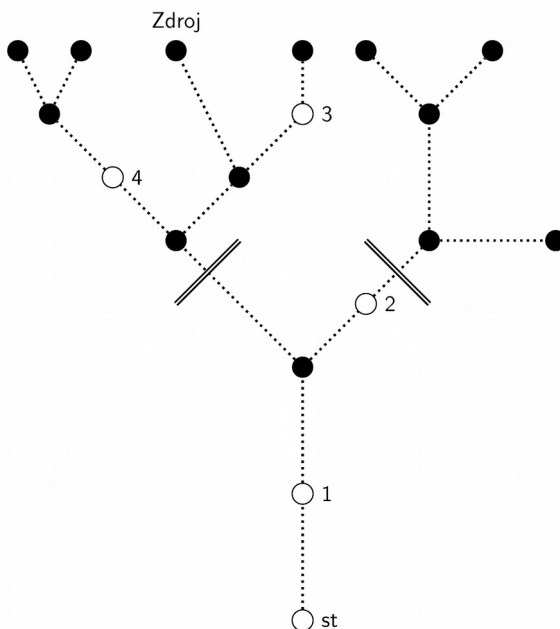
V posledním řádku tabulky jsou výsledky simulace při nastavení  $A = 1$ ,  $B = 99$ , přičemž očekávaná hodnota délky hry je 99 kol. To odpovídá 92% kvantilu rozdělení, získaného simulací s 1000 hráči.

Ruinování hráče představuje jednoduchý a dobře popsany problém, kde řada otázek typu „*Jaká je nejkratší možná doba hry?*“, „*Jaká je pravděpodobnost, že hra skončí do pěti kol?*“, „*O kolik větší jmění musí hráč mít, aby kompenzoval o 10 % nižší šanci výhry?*“ může být zodpovězena analyticky a simulace nejsou potřeba. V případě složitějších zadání, kdy se pravděpodobnost výhry mění na základě aktuálního jmění, mění se výška sázky apod. mohou být simulace užitečné a díky paralelnosti výpočtu představují P systémy zajímavou alternativu k jiným výpočetním metodám, založeným na opakované simulaci hry jednoho hráče.

Reálné systémy, v nichž jsou principy náhodné procházky (ať již férové či neférové) zřídka odpovídají jednoduchému modelu, kdy jsou jak pravděpodobnosti  $p$ ,  $q$ , tak i velikost sázky konstantní po celou dobu výpočtu. Návrh P systémů je variabilní a umožňuje vytvořit model systému, který tyto prvky obsahuje.

### Šíření znečištění říční sítí

Je zkoumána hypotetická situace, kdy dojde k bodovému znečištění vodního toku. Pro simulaci je použita membránová struktura, zobrazená na obrázku 5. Jako *Zdroj* je označena membrána, v níž dojde ke znečištění. Znečištění je reprezentováno pomocí objektu *cont*. Je zkoumána doba, kterou trvá, než znečištění opustí systém (než budou všechny objekty *cont* v membráně *Ústí*) a je sledováno, jakých koncentrací znečištění nabývá v jednotlivých membránách během výpočtu.



Obr 5. P systém pro modelování znečištění říční sítě

Navržený model nepracuje se signalizací (všechny membrány jsou otevřené) ani s katalyzátory. Model obsahuje pravidla popisující dva typy operací:

- Setrvání v membráně: Toto pravidlo se nachází ve všech membránách s výjimkou *Ústí* a popisuje situaci, kdy znečištění setrvává v aktuální membráně (ať již se jedná o rybník, soutok nebo pramen). S pravidlem je asociována pravděpodobnost  $p$ .
- Opuštění membrány: Toto pravidlo se nachází ve všech membránách kromě *Ústí* a přesune objekt *cont* do některé ze sousedních membrán, pokud mezi dvojicí membrán existuje synapse. Vykonání trvá  $t$  časových jednotek, s pravidlem je asociována pravděpodobnost  $p$ . Je vyžadována pouze existence synapse, na jejím stavu nezáleží (model předpokládá, že jsou všechny synapse po celou dobu výpočtu otevřené).

Membrána *Ústí* neobsahuje žádná pravidla – pokud se objekt *cont* dostane do této membrány, již zde zůstane. Oproti výše představeným modelům dochází k výpočtu tak, jak je definovaný

v klasické variantě P systémů. V okamžiku, kdy jsou všechny objekty *cont* přítomny v membráně *Ústí*, není možné použít již žádné další pravidlo a výpočet končí. Předchozí simulace sledovaly stav systému v jednotlivých časových krocích, zde je sledován počet kroků, které je nutné provést než dojde k úspěšnému ukončení výpočtu.

Z popisu modelu je zřejmé, že je zjednodušením transportních P systémů se signalizací a stejný systém by bylo možné simulovat pomocí jiných variant P systémů, například generalizovaných komunikačních P systémů (Verlan et al., 2008).

Model obsahuje pouze tři parametry - pravděpodobnost setrvání v membráně ( $p_{here}$ ), pravděpodobnost transportu po proudu ( $p_{down}$ ) a pravděpodobnost transportu proti proudu ( $p_{up}$ ).

Je sledována doba výpočtu (doba, než všechny kontaminanty pustí systém) a též, zda se během výpočtu dostanou kontaminanty do membrán, označených jako *Rybník*.

Jsou simulovány tři scénáře, přičemž čas transportu mezi dvěma synapsemi je vždy 3 časové jednotky. Pro každý scénář je provedeno 100 simulací, jejichž výsledky jsou popsány níže.

**První scénář** předpokládá vysokou pravděpodobnost transportu po proudu ( $p_{down} = 10$ ), střední pravděpodobnost setrvání ( $p_{here} = 5$ ) a nízkou pravděpodobnost transportu proti proudu ( $p_{up} = 1$ ). U všech rybníků dojde ve více než polovině případů ke kontaminaci; doba, po kterou jsou znečišťující látky přítomny v systému se pohybuje v průměru okolo 50 časových kroků. Nejkratší cesta z membrány Zdroj do Ústí trvá 22 časových jednotek. Při tomto nastavení trvá „vyčištění“ systému od znečišťujících látek v průměru více než dvojnásobek nejkratší možné doby.

**Druhý scénář** předpokládá nízkou pravděpodobnost transportu po proudu ( $p_{down} = 2$ ), i setrvání ( $p_{here} = 2$ ) a nízkou pravděpodobnost transportu proti proudu ( $p_{up} = 1$ ). Při tomto nastavení, které předpokládá malý rozdíl mezi pravděpodobnostmi trvá v průměru téměř 300 časových jednotek, než dojde k ukončení výpočtu a odstranění všech znečišťujících látek. Zároveň téměř při všech simulacích došlo ke kontaminaci všech rybníků (jediný rybník 2 nebyl kontaminován ve všech bžích simulátoru).

**Třetí scénář** předpokládá velmi vysokou pravděpodobnost transportu po proudu ( $p_{down} = 100$ ), střední pravděpodobnost setrvání ( $p_{here} = 10$ ) a nízkou pravděpodobnost transportu proti proudu ( $p_{up} = 1$ ). Ve všech bžích simulace došlo k odstranění

kontaminujících látek ze systému za méně než jeden a půl násobek nejkratší možné doby. Polovina z kontaminujících látek byla odstraněna za dobu jen o jednu jednotku delší, než je nejkratší možná doba. Jediný rybník, kde došlo ke kontaminaci, je rybník 1, který se však nachází na cestě ze Zdroje znečištění do Ústí a kontaminován být musí. U žádného z ostatních rybníků nedošlo v žádné ze simulací ke kontaminaci.

Transport látek říčním proudem je proces, který je ze své podstaty mechanický. Přenášený materiál nečiní rozhodnutí a nemůže aktivně ovlivnit svou cestu sítí. Z tohoto pohledu se jedná o ideální proces pro modelování v P systémech (zde představená simulace uvažovala i možnost transportu látek proti proudu, ovšem čistě kvůli demonstraci použití modelu). Kromě šíření znečištění je možné modelovat například sedimentaci či erozi, tání nebo naopak vyčerpávání vodních zásob. Detailní modelování těchto jevů sice P systémy neumožňují (pro všechny výše zmíněné jevy existují podrobné fyzikální modely), nicméně na úrovni celých povodí nebo i vyšších geografických celků můžou P systémy poskytovat přiměřenou míru generalizace a zároveň být dostatečně robustním a parametrizovatelným modelem.

### **Šíření invazivního druhu říční sítí**

Simulace demonstruje použití transportních P systémů pro modelování šíření invazivních druhů. Modelovaný systém je navržen jako říční síť obsahující rybníky. Je použita stejná membránová struktura jako v předchozí simulaci. Jako *Rybníky* jsou na obrázku 5 označeny membrány s čísly 1 až 4. Membrána *Ústí* je koncovým bodem systému.

Černě označené membrány reprezentují soutoky a prameny. V *Ústí* je na počátku výpočtu 250 objektů a, reprezentujících organismus. Pomocí synapsí, které jsou mezi každými sousedícími membrány vždy v obou směrech, dochází k pohybu organismu sítí. Všechny membrány obsahují pravidla pro transport sítí v obou směrech i pravidlo, popisující setrvání. Pravidla mají asociované pravděpodobnosti, přičemž pravděpodobnost pohybu po proudu a proti proudu se liší. Membrány dále obsahují katalyzátory, které reprezentují potravní zdroje. Pravidlo popisující setrvání v membráně je podmíněno přítomností potravy – katalyzátoru. Signalizace je použita k simulaci zámružky řeky. V obrázku 5 reprezentuje dvojitá čára kolmá k synapsi hranici zámružky řeky. Synapsím nad hranicí zámružky je periodicky měněn stav, přičemž stav  $s^-$  má význam zamrzlé řeky. Pokud je synapse v uzavřeném (zamrzlém) stavu, není ji možné použít k transportu objektů.

Synapse pod hranicí zámru zůstávají v otevřeném stavu  $s^+$  po celou dobu výpočtu.

Tab3: Nastavení parametrů simulace šíření invazivního druhu

Parametr	Hodnota
Počet organismů v Ústí (v počátku výpočtu)	250
Množství potravy v Ústí	500
Množství potravy v Rybnících	50
Množství potravy v soutocích a pramenech	5
Pravděpodobnost setrvání v membráně	10 / 10 / 100
Pravděpodobnost opuštění membrány	5 / 5 / 50
Pravděpodobnost úhynu	0 / 1 / 1
Čas pro překonání synapse	2
Čas po který je synapse otevřená (splavná)	9
Čas po který je synapse uzavřená (zamrzlá)	3

Jsou porovnány tři nastavení systému, lišící se v pravděpodobnosti úhynu. Pravděpodobnost setrvání v membráně je vždy dvojnásobná oproti pravděpodobnosti opuštění membrány. V jednom případě je úhyn vyloučen ( $p_{\text{úhyn}} = 0$ ), v jednom případě je jeho pravděpodobnost desetinná oproti pravděpodobnosti setrvání ( $p_{\text{úhyn}} = 1$ ,  $p_{\text{setrvání}} = 10$ ) a v poslední variantě nastavení je setinová ( $p_{\text{úhyn}} = 1$ ,  $p_{\text{setrvání}} = 100$ ) v porovnání s pravděpodobností setrvání.

Membrány obsahují čtyři typy pravidel:

- Setrvání v membráně: Toto pravidlo se nachází ve všech membránách a popisuje situaci, kdy živočich setravá ve stávající membráně. K vykonání pravidla je nutné, aby byl v membráně přítomen katalyzátor  $k$ , reprezentující potravu. S pravidlem je asociována pravděpodobnost  $p$ .
- Opuštění membrány: Toto pravidlo se nachází ve všech membránách a přesune organismus do některé ze sousedních membrán, pokud existuje otevřená synapse. Vykonání trvá  $t$  časových jednotek, s pravidlem je asociována pravděpodobnost  $p$ .
- Úhyn: Toto pravidlo se nachází ve všech membránách a odstraní jednu instanci objektu  $a$  z membrány, čímž je simulován úhyn organismu.
- Zamrznutí a rozmrznutí řeky: Tato pravidla se nachází ve všech membránách, které jsou „nad hranicí zámru“ nebo s těmito membránami přímo sousedí. Délka trvání



zamrznutí a rozmrznutí řeky se liší. K vykonání pravidla není nutné, aby se v membráně nacházel jakýkoli objekt.

Všechna tři nastavení systému byla modelována 100 krát po dobu 100 kroků. V membráně *Ústí* při všech nastaveních systému dojde k rychlému poklesu počtu objektů (katalyzátory nejsou počítány). V případě, že je modelován úhyn ( $p_{\text{úhyn}} > 0$ ), se počty limitně blíží nule. V případě, že není modelován, se na konci simulace (po 100 krocích) v membráně nachází okolo 20 jedinců, ačkoli je v membráně přítomno 500 katalyzátorů. Pokles počtu objektů je hyperbolický.

V membráně označené 1 (reprezentující rybník) dojde k prudkému nárůstu počtu jedinců na hodnotu cca 100, po kterém opět následuje pokles, jehož tvar se blíží hyperbole. Snižování počtu je pomalejší než v membráně *Ústí*, přičemž pokud je počítáno s úhynem, dojde během 100 kroků simulace k vymření a v membráně nejsou přítomny žádné objekty. Pokud s úhynem není počítáno, nachází se po 100 krocích v membráně v průměru 15 objektů, ačkoli membrána obsahuje 50 katalyzátorů, sloužících jako potravní zdroje.

Membrána 2 (reprezentující rybník) je od *Ústí* vzdálenější (cesta přes 3 synapse) a i když vývoj početnosti vykazuje podobný trend jako u membrán *Ústí* a 1, je maximální počet jedinců v systému nižší a pokles počtu pomalejší. V membráně 1 nedochází k zamrznutí, v membráně 2 dochází k zamrznutí jedné synapse ze dvou, které do membrány 2 směřují. Zamrznutí se projevuje cyklickými výkyvy, korespondujícími s periodami zámrazu. U všech zatím zkoumaných membrán byl pozorován stejný trend prudkého nárůstu počtu jedinců a následného poklesu, který má tvar hyperboly. Čím dále od *Ústí* se membrána nachází, tím je maximum počtu objektů v membráně nižší a tím je pokles pozvolnější. Pokud je počítáno s úhynem, dochází vždy k vymření v dané membráně.

Membrány 3 a 4, které rovněž reprezentují rybníky, ale nacházející se v části říční sítě, která zamrzá, ukazují odlišný trend v počtu jedinců. V žádné z membrán nedochází k prudkému nárůstu počtu jedinců, naopak nárůst je pozvolný a bez výrazného vrcholu. V případě, že není počítáno s úhynem, se počet jedinců v membráně ustálí okolo hodnoty 10 jedinců, ačkoli membrány obsahují potravní zdroje pro 50 jedinců. Pokud systém obsahuje pravidla pro úhyn, dojde ke snižování počtu jedinců až k nule, přičemž čím větší je relativní pravděpodobnost úhynu, tím menších

maxim počet jedinců dosahuje. Maximální počet jedinců se rovněž snižuje se vzdáleností od *Ústí*.

Výsledky ukazují, že čím vzdálenější je membrána od *Ústí*, které představuje zdroj objektů, tím pomalejší je nárůst počtu jedinců, tím nižší je maximum, kterého je v membráně dosaženo a tím pomalejší je pokles po dosažení maxima.

V membránách, do nichž vedou zamrzající úseky říční sítě jsou pozorovány cyklické výkyvy, korespondující s periodou zamrzání, v membránách v nezamrzající části říční sítě tyto výkyvy patrně nejsou. Ačkoli membrány reprezentující rybníky obsahují katalyzátory, umožňující setrvání 50 jedinců, hodnota na níž se počet jedinců ustálí, je nižší. Namísto koncentrace do rybníků je pozorována disperze, která směřuje k rovnoměrnému rozmístění objektů ve všech membránách bez ohledu na potravní zdroje. V případě, že je v systému přítomno 250 jedinců a systém obsahuje 18 membrán, vychází v průměru 14 jedinců na membránu. Při simulaci 1000 kroků dojde k rozptylu a všechny membrány obsahují v průměru okolo 14 jedinců, přičemž počty v membránách nezůstávají konstantní, ale kolísají v rozmezí cca 3 až 20 objektů v každé membráně. To platí i pro membrány reprezentující prameny a soutoky. V případě, že existují pravidla popisující úhyn, dochází ve všech membránách k vymření. Čím větší je relativní pravděpodobnost úhynu v porovnání s pravděpodobností setrvání, tím menších maxim dosahuje počet objektů v jednotlivých membránách, přičemž čím dále se membrána nachází od zdroje (*Ústí*), tím je tento efekt výraznější.

Transportní P systémy neumožňují vytvořit pravidla popisující růst, jediným zdrojem objektů v systému je vnější prostředí, což limituje použití pro věrohodnější modelování populací, v principu však není problém model rozšířit tak, aby na levé i pravé straně pravidla mohlo být více než jen jeden objekt (a tudíž model byl schopen reprezentovat růst populací). Takový model není v této práci podrobněji zkoumán, neboť primárním cílem práce je zkoumat transport v P systémech a nikoli vývoj populací. Setrvání v membráně popisuje pravidlo, které obsahuje katalyzátor, reprezentující potravu, zatímco pravidlo popisující odchod z membrány katalyzátor neobsahuje. Namísto katalyzátoru  $k$  by zde bylo možné použít inhibitor  $-k$ , jehož význam by byl „*pokud není přítomen objekt  $k$ , vykonaj pravidlo*“. Tento způsob zápisu lépe popisuje pohyb živočichů v závislosti na dostatku potravních zdrojů než pravděpodobnosti asociované s pravidly, vyžadoval by však úpravy navrženého modelu, který s inhibitory nepracuje. Potenciálním problémem při použití inhibitorů je přílišný

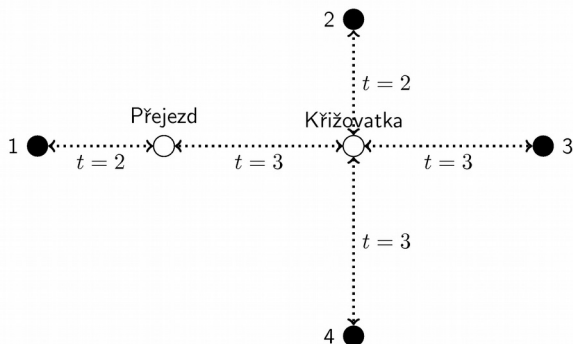
determinismus systému; pokud existují pouze dvě pravidla, z nichž jedno popisuje setrvání a vyžaduje přítomnost katalyzátoru a druhé popisuje opuštění membrány a vyžaduje jeho absenci, pak jsou pravidla vykonávána přísně deterministicky v závislosti na počtu katalyzátorů v membráně a samovolné opuštění membrány navzdory přítomnosti potravních zdrojů není možné modelovat.

Z výše řečeného vyplývá, že model není zcela vhodný k modelování populačního růstu a/nebo metapopulací (prostorově oddělených populací), nicméně může sloužit jako základ, který je rozšířen pro účely biologického modelování. Jako nezbytné se jeví zahrnout do modelu pravidla umožňující reprezentaci růstu populace a detailnější popis mechanismu disperze populace. Dynamický výpočet pravděpodobnosti opuštění membrány na základě aktuální konfigurace membrány je jednou z možností, použitou například v práci (Pérez-Jiménez a Romero-Campero, 2006); druhou možností je nepracovat s pravděpodobností, ale popsat disperzi pravidly, pracujícími s katalyzátory, promotory/inhibitory a dalšími prvky, které omezují nedeterminismus systému.

### **Vliv kapacity křižovatky na rozpuštění fronty**

Simulace uvažuje systém, vyobrazený na obrázku 6: Existují 4 body (brány), sloužící jako bílé a černé díry, křižovatka a železniční přejezd. Brána je ve skutečnosti dvěma membránami, z nichž jedna slouží jako černá díra a jedna jako bílá díra. Tyto dvě membrány nejsou spojeny synapsí, existují pouze synapse vedoucí z bílé díry do sousední membrány a ze sousední membrány do černé díry. Tři brány jsou spojeny přímo s křižovatkou. Čtvrtá brána je spojena s přejezdem, který je dále spojen s křižovatkou.

*Obr 6. Schéma membránové sítě pro simulaci rozpuštění fronty*



*Křižovatka* má danou kapacitu  $k$ , v bílých děrách jsou generovány vozidla každou časovou jednotku. *Přejezd* je reprezentován jako křižovatka bez signalizace s neomezenou kapacitou (vozidla mohou membránu opouštět, pokud existuje otevřená synapse, přítomnost katalyzátoru není nutná). Čas nezbytný k překonání synapsí je zobrazen na obrázku 6 (označen jako  $t$ ). Membrána označená jako *Přejezd* je křižovatkou, která neobsahuje signalizaci a je synapsemi spojená jen se dvěma dalšími membránami. Všechny synapse vedoucí do a z přejezdu jsou na počátku simulace otevřeny. Průjezd přejezdem trvá dvě časové jednotky (v prvním kroku je do membrány přiveden objekt  $veh$ , který se změní na objekt  $veh(in_i)$  a ve druhém kroku membránu opouští). Systém se tedy chová stejně, jako kdyby byly membrána 1 a *Křižovatka* spojeny přímo synapsí.

V systému jsou definovány události, které simulují průjezd vlaku přes přejezd. Události nejprve uzavřou (v čase  $t = 23$ ) a pak otevřou (v čase  $t = 30$ ) synapse vedoucí z přejezdu do sousedních membrán (*Křižovatka* a brána 1). Membrána *Přejezd* je v systému modelována jako křižovatka. To však není správná reprezentace s ohledem na realističnost výběru cest vozidel. Jelikož je předpokládána stejná pravděpodobnost výběru pro všechny synapse, vozidla, která se ocitnou v membráně *Přejezd* budou s 50% pravděpodobností přemístěny do membrány 1 a s 50% pravděpodobností do membrány *Křižovatka*.

Pokud například v jednom kroku z membrány *Křižovatka* přijde 20 vozidel a z membrány 1 přijde 5 vozidel, očekávané chování systému je takové, že 20 vozidel bude pokračovat do membrány 1 a 5 vozidel bude pokračovat do membrány *Křižovatka*. Ve skutečnosti však dojde k tomu, že (v průměru) každou ze synapsí

bude přemístěna polovina objektů. Do membrány Křižovatka bude směřovat 12,5 namísto 5 objektů. Membrána Křižovatka obsahuje pravidla, měnící stav synapsí, které z membrány vedou do sousedních membrán. Na počátku výpočtu jsou synapse (Křižovatka, Přejezd) a (Křižovatka, 3) otevřeny a synapse (Křižovatka, 1) a (Křižovatka, 2) uzavřeny. Stav synapsí se mění po dvou krocích. V každém okamžiku jsou otevřeny dvě synapse vedoucí z Křižovatky, zatímco dvě jsou uzavřeny.

Intenzita dopravy je na všech komunikacích totožná, tudíž pravděpodobnost výběru pravidla je stejná pro všechny cesty (pro všechny membrány, dostupné z membrány Křižovatka).

Jsou porovnány 4 varianty nastavení systému:

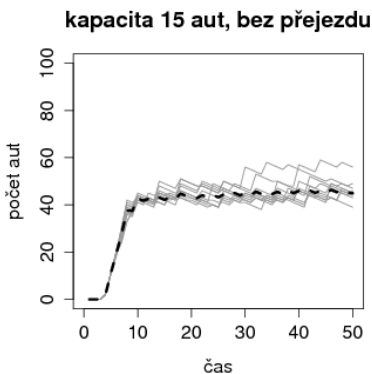
Tab: 4: Nastavení parametrů simulace rozpuštění fronty

Přejezd vlaku	Kapacita křižovatky
Ne	15
Ano	15
Ano	20
Ano	25

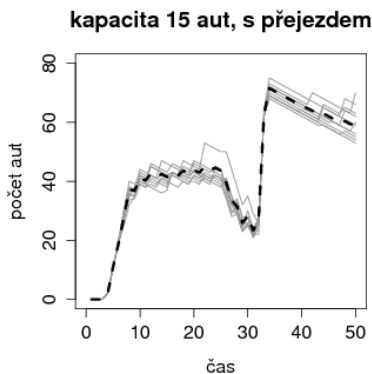
První dvě porovnávané situace se liší tím, zda přes přejezd přejede vlak. Pokud ano, dojde po dobu 6 kroků k uzavření synapsí vedoucích z membrány Přejezd. Poté, co jsou synapse znovu otevřeny již zůstanou otevřené. Druhým parametrem, který se mezi jednotlivými simulacemi liší, je kapacita křižovatky, vyjádřená počtem vozidel, která mohou křižovatkou projít za jednu časovou jednotku. V membránách 1 a 3 je každou časovou jednotku generováno 5 vozidel, v membránách 2 a 4 jsou generována 2 vozidla. Počet generovaných vozidel je konstantní po celou dobu výpočtu. Zde prezentované výsledky ukazují simulaci po dobu 50 časových jednotek. Pro každou ze čtyř porovnávaných situací je provedeno 50 simulací. Na obrázcích 5 až 8 jsou zaznačeny počty vozidel v membráně Křižovatka v jednotlivých simulacích (světle šedá barva) a jejich průměr (černá čárkovaná linie). Zobrazené počty ukazují všechny objekty reprezentující vozidla, tedy jak objekty *veh*, tak *veh(in)*. Katalyzátory nejsou v počtu zahrnuty.

Ačkoli počet vozidel, která jsou v každém kroku generována je konstantní a stav signalizace je měněn ve stále stejných časových krocích, je systém nedeterministický. Vozidlu, které přijde do membrány je umožněno, aby se v dalším kroku vrátilo do

membrány, z níž přišlo. Jinak řečeno, mezi jednotlivými vozidly se nerozlišuje a jsou-li v membráně přítomny jako objekt *veh*, není již možné zjistit, ze které membrány do současné membrány přišly a sledovat historii jejich pohybu v systému. Náhodný výběr membrán, do nichž budou objekty přemísťovány (pravděpodobnosti výběru všech synapsí jsou totožné) přináší do systému nedeterminismus.



*Obr. 7: Kapacita 15 vozidel za časovou jednotku, bez přejezdu*

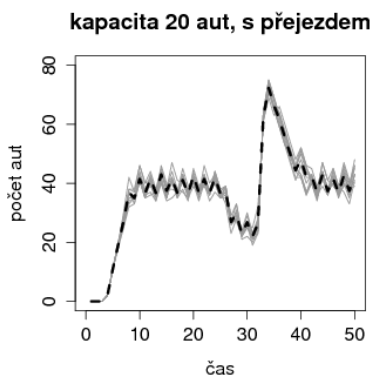


*Obr. 8: Kapacita 15 vozidel za časovou jednotku, s přejezdem*

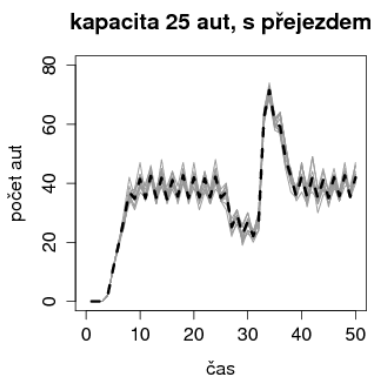
Z porovnání obrázků 7 a 8 je patrné, že pokud dojde k přejezdu vlaku – dočasného uzavření synapsí vedoucích z membrány Přejezd – je nejprve snížen počet příjezdějících vozidel, který je následně kompenzován náhlým přísunem téměř 50 vozidel, která jsou v několika časových krocích z křižovatky odvedena. Rychlost mizení fronty závisí na kapacitě křižovatky. V případě kapacity 15 vozidel dochází k rozpouštění fronty velmi pomalu, zatímco při kapacitě 20 vozidel zmizí fronta za 10 časových jednotek. Se zvyšující se kapacitou (25 vozidel) již k výraznému urychlení odbavení vozidel v křižovatce nedochází. Po zmizení fronty osciluje počet vozidel v křižovatce okolo 40 (započítána jsou všechna vozidla).

Provedená simulace nepřináší výsledky, které by byly v některém ohledu překvapivé, nicméně demonstruje schopnost navrženého modelu P systémů realisticky modelovat dopravní proud a to včetně situací, které nelze v rámci některých modelovacích přístupů jednoduše řešit. Použití událostí je rozšířením P systémů,

keré lze jednoduše použít pro modelování náhodných, nepravidelných nebo málo četných událostí.



*Obr. 9: Kapacita 20 vozidel za časovou jednotku, s přejezdem*



*Obr. 10: Kapacita 25 vozidel za časovou jednotku, s přejezdem*

## 5. Transportní P systém s vozidly

### Formální definice:

Transportní P systém s vozidly je konstrukt

$\Pi = (O, C, \mu, W_{m1}, \dots, W_{mk}, W_{v1}, \dots, W_{vl}, syn, R, R_p, i_o)$ , kde

- $O$  je abeceda objektů,
- $C \subset O$  je abeceda katalyzátorů,
- $\mu = \{m1, \dots, mk, v1, \dots, vl\}$  je množina membrán s  $k$  stanicemi a  $l$  vozidly. Stanice se nachází přímo ve vnějším prostředí, membrány reprezentující vozidla jsou elementární a nachází se ve stanicích,
- $W_{m1}, \dots, W_{mk}, W_{v1}, \dots, W_{vl}$  jsou multimnožiny objektů, nacházejících se na počátku výpočtu v jednotlivých stanicích  $m1, \dots, mk$  a vozidlech  $v1, \dots, vl$ ,
- $syn \subseteq \{(i, j, t) \mid j \in \{m1, \dots, mk\}, i \prec j, t \in N^+\}$  je množina synapsí mezi membránami. Každá synapse obsahuje informaci o času  $t$ ,
- $R_{m1}, \dots, R_{mk}, R_{v1}, \dots, R_{vl}$  je množina pravidel asociovaných s jednotlivými membránami systému,
- $R_p$  je množina visících pravidel,
- $i_o$  je buď jeden z regionů  $m1, \dots, mk, v1, \dots, vl$  a pak je tento region výstupním regionem systému, nebo je  $O$  a pak je výsledek výpočtu uložen ve vnějším prostředí systému.

Pravidla mají následující tvar:

- $(xk, in; y, out)$  je antiportní pravidlo, kde  $x, y \in O$  jsou objekty (nikoli multimnožiny objektů) a  $k \in C$  je katalyzátor. Pravidlo popisuje situaci, kdy objekt  $x$  je přemístěn dovnitř membrány, zatímco objekt  $y$  ji ve stejném kroku opustí. Objekt  $k$  slouží katalyzátor a je použit standardním způsobem.
- $(xk, ind)$  je uniportní pravidlo, kde  $x \in O$  je objekt a  $ind \in \{in, out, here\}$  je indikátor cíle. Pokud  $ind = in$ , objekt je přesunut do membrány vnitřní, pokud  $ind = out$ , objekt je přesunut do membrány rodičovské (pokud je současná membrána vnější membránou, je objekt přesunut do vnějšího prostředí) a pokud  $ind = here$ , pak objekt zůstává v současné membráně. Objekt  $k \in C$  slouží jako katalyzátor a je použit standardním způsobem.
- $([j], in\ j, t)$  je pravidlo, popisující přemístění vnitřní membrány (označené jako  $[j]$ ) do membrány  $j$ . Podmínkou je existence synapse mezi membránou, v níž je pravidlo umístěno a membránou  $j$ . Přemístění membrány zabere  $t$  časových jednotek.
- $([j], \delta)$  je pravidlo, popisující rozpuštění vnitřní membrány.



- $(I, \varepsilon)$  je pravidlo, popisující vytvoření membrány uvnitř současné membrány.

## Popis modelu

Pravidla představeného modelu navazují na komunikační P systémy, P systémy s aktivními membránami, P systémy s mobilními membránami a P systémy s nosiči. Od všech těchto modelů se však model transportních P systémů s nosiči v některých aspektech odlišuje. Zde použitý zápis využívá symbolů  $\delta$  a  $\varepsilon$  pro označení operací rozpuštění, respektive vytvoření membrány.

Zatímco v modelu transportních P systémů byly objekty transportovány přímo mezi membránami, které se nacházely ve vnějším prostředí, zde jsou k transportu použity membrány, které samy mění polohu a přemísťují se. Existují tedy dva druhy membrán - membrány umístěné v uzlech sítě (stanice), kterými se do a ze systému dostávají objekty a membrány mobilní, které jsou mezi těmito membránami transportovány. Mobilní membrány slouží jako vozidla (nosiče), která umožňují transport objektů. Vozidla mohou být ve stanicích vytvářena a mohou v nich zanikat, zatímco stanice jsou pevně dané a po celou dobu výpočtu neměnné.

Katalyzátory jsou nutné s ohledem na práci s kapacitou vozidel. Katalyzátory se nacházejí v mobilních membránách a jejich počet odpovídá maximálnímu počtu objektů, které mohou být vozidlem transportovány.

Vzhledem k tomu, že mobilní membrány vznikají a jsou rozpouštěny, je nutné, aby katalyzátory byly v membránách přítomny v okamžiku vytvoření membrány. V případě rozpouštění membrány jsou všechny objekty v ní přítomné přesunuty do membrány vnější; pokud taková membrána neexistuje, tak do vnějšího prostředí.

Synapse, stejně jako v transportních P systémech, obsahují informaci o čase, který trvá vykonání cesty skrz synapsi. Obdobně jako v předchozím modelu jsou použita visící pravidla, přičemž algoritmus aplikace visících pravidel zůstává stejný. Jediná pravidla, která obsahují informaci o čase, jsou pravidla popisující přemístění mobilních membrán.

Navržený model obsahuje membrány ve dvou úrovních. Ve vnějším prostředí jsou obsaženy membrány reprezentující stanice a v nich jsou obsaženy membrány reprezentující vozidla. Vnější prostředí nikdy nekomunikuje přímo s vozidly. Bylo by tedy možné systém navrhnout tak, že všechna pravidla by byla obsažena v membránách - stanicích - a tato pravidla by popisovala jak

transport objektů z vnějšího prostředí do stanic, tak ze stanic do vozidel a naopak. Takový systém by byl dostatečný za předpokladu, že přepravované objekty cestují vždy pouze mezi dvěma sousedními membránami. To však nemusí být pravda - objekt může vstoupit do mobilní membrány a setrvat v ní libovolně dlouhou dobu. Musí tedy existovat pravidlo, které popisuje setrvání objektu v mobilní membráně. Tvar tohoto pravidla je  $(xk, here)$  a opisuje setrvání objektu  $x$  v současné membráně za předpokladu, že je v ní přítomen katalyzátor  $k$ .

Stejně jako v případě P systémů se symportem/antiportem jsou přítomná pravidla komunikační. Objekty jsou v rámci systému přemísťovány, ale nemění se jejich množství. Může se však měnit jejich typ (objekt  $u$  se během transportu může změnit v objekt  $v$ ) Jediný způsob, jak změnit množství objektu  $u$  v systému, je přivést jeho další kopie z vnějšího prostředí.

Mobilní membrány, pokud jsou vytvářeny, musí být vytvářeny včetně objektů (katalyzátorů) a pravidel, což zde představený model odlišuje od P systému s mobilními membránami (Krishna a Păun, 2005; Aman a Ciobanu, 2008).

Model transportních P systémů s nosiči se nejvíce blíží P systémům s nosiči (Martín-Vide et al., 2002b), proto i podobnost názvů. P systémy s nosiči používají podobný princip pohybu objektů - vytvoření membrány, v níž jsou objekty přesunuty a její následné rozpuštění. Nosiče však slouží pouze k transportu „vertikálně“ - do membrán vnitřních nebo vnějších. Zde představený model přemísťuje nosiče „horizontálně“ - mezi membránami na stejné úrovni, které jsou spojeny synapsemi.

Druhý podstatný rozdíl spočívá v tom, že nosič není po vstupu do další membrány rozpuštěn, ale dále existuje. V P systémech s nosiči (Martín-Vide et al., 2002b) vždy nosič po transportu mezi dvojicí membrán zaniká, ve zde představených transportních P systémech s nosiči dále existuje a může se membránovou strukturou pohybovat po celou dobu výpočtu. Z tohoto rozšíření plyne další významná vlastnost - objekty mohou v nosičích zůstat po libovolně dlouhou dobu. Objekt může být zapouzdřen do nosiče v membráně  $i$ , opustit nosič v membráně  $k$  a mezitím být přemístěn libovolným počtem dalších membrán.

## 6. Výsledky

Tato práce představuje dva modely transportu v P systémech. Oba simulují transport objektů v předem dané síti, která je statická. Modely nejsou schopny zachytit transport popsaný například vektory pohybu (směr, vzdálenost), dokáží pouze přesouvat objekty v síti, jejíž uzly mohou být lokalizovány v prostoru. Pro simulaci jevů, dějících se ve spojitém prostoru je možné uspořádat síť membrán do pravidelné mřížky s konstantní délkou synapse mezi sousedícími membránami. Tento princip práce s prostorem byl použit v práci (Barbuti et al., 2011) jako takzvané prostorové P systémy.

Zde navržené modely aspirují modelovat jiný druh transportu. Představené modely předpokládají, že jediné lokalizované prvky systému jsou membrány. Pro objekty platí pravidlo, že jsou v rámci membrán neuspořádány a jsou vzájemně zaměnitelné. Membrány reprezentují v prostoru lokalizované entity reálného světa, jejich absolutní poloha v rámci daného souřadného systému se výpočtu neúčastní, může však být atributem membrán. Z hlediska reprezentace prostoru v geografických informačních systémech odpovídají navržené varianty P systému vektorovému modelu dat, zatímco prostorové P systémy odpovídají rastrovému modelu dat.

Oba zde navržené modely jsou postaveny na podobném schématu, kdy membrány jsou rozmístěny v grafu a spojeny synapsami, podobně jako v tkáňovitých P systémech (Martín-Vide et al., 2002a). Existence synapse vyjadřuje sousedství membrán (možnost přímo cestovat mezi danou dvojicí). Ačkoli při návrhu modelu bylo uvažováno, že synapse představují komunikace spojující místa v síti, nemusí platit, že synapse reprezentují entity reálného světa. Synapse nejsou fyzicky existujícími komunikačními cestami (silnicí, vodním tokem, potrubím apod.), ale pouze mentálním konstruktem, který stanovuje, že mezi dvěma membránami lze přímo cestovat. Vzdálenost mezi membránami může být vyjádřena pomocí pravděpodobnosti nebo času. V případě modelování například disperze populace může pravděpodobnost synapse být inverzní hodnotou vzdáleností, tudíž vzdálenější membrány budou mít nižší pravděpodobnosti výběru. Vhodnější způsob vyjádření vzdálenosti je však pomocí času, který transport mezi dvojicí membrán zabere. Například pro simulaci transportu vozidel v silniční síti je tato reprezentace dostačující, pro simulaci dálkové migrace živočichů zřejmě nikoli. Způsob, jak v rámci P systémů zahrnou vzdálenost (mezi membránami či

objekty) přímo do výpočtu nebyla zatím v literatuře publikována. Z pohledu P systémů je umístění v prostoru pouze atributem jednotlivých prvků systému, nikoli parametrem hrající roli při výpočtu.

Ze své podstaty jsou P systémy diskrétní a transport je realizován na úrovni jednotlivých objektů. Čas transportu, který je pro simulování reálných jevů zásadní, probíhá v předem daných časových jednotkách. Použití časovaných P systémů (Cavaliere a Zandron, 2006; Barbuti et al., 2008) umožňuje přiřadit pravidlům čas vykonání, který musí být násobkem základní časové jednotky. Oba modely navazují na P systémy se symportem/antiportem (Păun a Păun, 2002) a generalizované P systémy (Verlan et al., 2008), neboť pravidla, která jsou použita, jsou pravidly komunikačními. Dochází pouze k přesunu objektů v rámci systému, nikoli k vytváření nových objektů. Nové objekty mohou do systému vstupovat pouze z vnějšího prostředí. Kromě přesunu v rámci systému mohou objekty měnit svou kvalitu ( $u \rightarrow v$ ), nikoli však svou kvantitu ( $u \rightarrow uu$  není povoleno).

Model transportních P systému se signalizací, kapacitami a nosiči simuluje transport individuálních objektů v síti. Jsou využívány principy časovaných pravidel (Cavaliere a Zandron, 2006), katalyzátorů a stavů synapsí (podobně jako u tkáňovitých P systémů se stavy komunikačních kanálů (Freund et al., 2004)). Inspirací při návrhu modelu byl pohyb vozidel v dopravní síti se signalizovanými křižovatkami. Každá z membrán v systému reprezentuje buď křižovátku nebo vstupní a výstupní bod systému. Mezi kteroukoli dvojicí membrán může existovat nejvýše jedna synapse v každém směru. Existence více synapsí v některém ze směrů by vyžadovala změnu definice modelu. Synapse mohou nabývat právě jeden z předem definované množiny stavů, přičemž aplikace pravidel může být omezena existencí synapse v daném stavu. Ve všech příkladech použití zmíněných v této práci jsou však použity pouze dva stavy -  $s^+$  a  $s^-$ , reprezentující otevřený a zavřený komunikační kanál. Synapse zde v podstatě plní roli promotora (Ionescu a Sburlan, 2004), protože jejich přítomnost (v některém ze stavů) může být podmínkou vykonání pravidla. Objekty v systému jsou rozlišeny na běžné objekty a katalyzátory. Katalyzátory v rámci systému nemění své množství ani pozici a umožňují aplikaci pravidel. Katalyzátory v navrženém modelu reprezentují kapacitu křižovatek - počet vozidel, která mohou křižovátkou za časovou jednotku projet, zatímco stavy synapsí reprezentují průjezdnost křižovátkou, tedy signalizaci (světelnou či jinou).

Nedeterminismus systému je omezen použitím pravděpodobností pro výběr pravidel, čímž model navazuje na práce (Obtulowicz a Păun, 2003; Pescini et al., 2006). Pravděpodobnost je použita na úrovni výběru pravidel, kdy každé pravidlo má zadánu pseudopravděpodobnost výběru. Pro účely simulování dopravy může být pseudopravděpodobnost určena na základě pozorovaných intenzit dopravy.

Model není navržen ke sledování cesty konkrétních objektů v síti. U objektů je evidována pouze jejich aktuální poloha a indikátor následující stanice, nikoli celá cesta. Výsledky výpočtu nelze použít k činění závěrů o délce či průběhu cesty, pouze ke sledování množství objektů, které se v daných časových okamžicích nacházejí v jednotlivých částech systému. Výjimku tvoří situace, kdy systém obsahuje jediný objekt, jak bylo ukázáno při simulaci náhodné procházky.

Inovativními rozšířeními modelu jsou tzv. události a visící synapse. Události jsou pravidly, jejichž aplikace je jednorázová a je vázána na konkrétní časový krok. Příkladem jejich použití je například nenadálé přerušování komunikace, odstranění objektů z konkrétní membrány nebo naopak vložení objektů do jedné nebo více membrán.

Visící synapse bylo nutné zavést z důvodu, že navržený model kombinuje variabilní délku vykonání pravidel se změnou stavů synapsí. Visící synapse označují ty synapse, jejichž stav je právě měněn některým z visících pravidel. Stav visících synapsí nemůže být měněn. Toto omezení zaručuje, že synapse budou vždy právě v jednom stavu. Pokud by neexistovalo, mohlo by dojít k situaci, kdy během jednoho výpočetního kroku má být stav synapse změněn do několika stavů.

Ačkoli hlavním objektem zájmu této práce je využití P systémů pro simulace a nikoli výpočty, je poskytnut důkaz, že představený model je Turingovsky kompletní. Pro model transportních P systémů se signalizací, kapacitami a událostmi je vytvořen softwarový nástroj v jazyce R, který je spolu s návodem k použití přílohou této práce a je dostupný ze stránek [https://github.com/janoskaz/trans\\_p\\_systems](https://github.com/janoskaz/trans_p_systems).

Pomocí softwarového nástroje jsou vytvořeny simulace, které demonstrují použití navrženého modelu k řešení jak ryze teoretických problémů, tak i k simulaci reálných systémů.

Model je použit k simulaci náhodné procházky a tzv. ruinování hráče, které z ní vychází. Simulace ukazují, že navržený model je

schopen simulovat náhodnou procházku, přičemž P systémy jsou obzvláště vhodné pro simulování náhodné procházky s omezeným prostorem (pohlující nebo odrážející stěny). Simulace ruinování hráče ukazují, že výsledky her simulovaných modelem odpovídají očekávaným hodnotám, vypočteným numericky. Simulace náhodné procházky a ruinování hráče nevyžadují použití stavů synapsí ani událostí.

Výhodou představeného modelu je nedeterminismus výpočtu. Pokud má být provedena simulace tisíce her, není nutné tisíckrát simulovat hru jednoho hráče; do počáteční membrány je vloženo tisíc hráčů a je simulována jedna hra.

Další simulace ukazují možnost použití modelu pro modelování šíření invazivního druhu v rámci říční sítě a simulování šíření znečištění. Stav synapsí je zde použit k reprezentaci zamrznání hladiny vodních toků, katalyzátory reprezentují dostupné potravní zdroje a události mohou být použity k simulaci jednorázových zásahů do systému – ať už je to vyhubení organismu v určité části systému, nebo naopak vložení populace do jedné z membrán.

Poslední simulace se zabývá transportem vozidel v rámci části silniční sítě a je ukázkou modelování procesů, k nimž byl model původně navržen. Události zde slouží k modelování přejezdu vlaku, který způsobí dočasnou neprůjezdnost synapse. Stav synapsí jsou použity k řízení křižovatky a kapacita určuje počet vozidel, která mohou křižovatkou během jedné časové jednotky projet.

Prezentované simulace ukazují poměrně široké použití modelu. Ačkoli byl model navržen na základě transportu vozidel v silniční síti, paradoxně vidím větší použití při modelování jevů, spadajících do oblasti fyzické geografie (sedimentace látek, jejich šíření v říční síti) nebo biogeografie (populační biologie). Silou představených modelů je práce s jednotlivými objekty, na rozdíl od práce s populací, popsanou sadou parametrů. Toto může být výhodou zejména při modelování vývoje málo početných systémů, například reliktních populací. Model lze rovněž použít pro matematické modelování, například náhodných procházek v grafu.

Druhý představený model se nazývá transportní P systémy s nosiči a k transportu objektů používá mobilních membrán. Blízce se podobá P systémům s nosiči (Martín-Vide et al., 2002b). Transportní P systémy s nosiči byly ve svém návrhu inspirovány transportem cestujících v prostředcích hromadné dopravy. Rozdílným prvkem oproti transportním P systémům je způsob přepravy objektů mezi membránami, který neprobíhá samostatně, ale uvnitř nosiče

- mobilní membrány, která je posílána z jedné membrány do druhé. Mobilní membrány mohou vznikat a zanikat a obsahují pravidla (tzn. nejsou to jen prázdné schránky). Kapacita vozidel je reprezentována pomocí katalyzátorů, jejichž počet v mobilní membráně odpovídá maximálnímu počtu objektů, které mohou být přepravovány.

Přizpůsobení navrženého modelu tak, aby věrohodně simuloval reálnou dopravu vyžaduje komplikované rozšíření, které efektivně vede k téměř úplnému omezení nedeterminismu systému. Nedeterminismus, společně s paralelismem jsou však základními vlastnostmi P systémů. Navržený model se proto k modelování reálných dopravních systémů příliš nehodí. Přeprava cestujících a osobní doprava obecně je tématem, které je dobře prozkoumáno a znalosti problematiky umožňují sestavení detailních modelů (Hoogendoorn a Bovy, 2001). P systémy se ukázaly vhodné spíše pro modelování systémů, které jsou inherentně nedeterministické. Osobní doprava je oproti tomu systémem, ve kterém kritickou roli hrají rozhodovací procesy. Ty nemohou být v rámci P systémů věrohodně reprezentovány. Rozhodování na úrovni vyhodnocování aktuální situace je procesem, který lze mnohem lépe modelovat například multiagentními systémy.

Zde navržené modely jsou proto vhodnější pro reprezentaci procesů, při nichž nedochází k vědomému rozhodování a umožňují provádět simulace s nastavením různých parametrů. P systémy byly navrženy jako model, který v sobě skloubí nedeterminismus a maximální paralelismus. Proto je vhodné použít je k modelování jevů, které ze své podstaty mají tyto vlastnosti. Jevy, které jsou deterministické (a osobní doprava mezi ně z velké části patří), nevyužívají plně vlastností P systémů a expresivnost, kterou P systémy poskytují, nemusí být v důsledku dostatečným důvodem k tomu, aby byl tento model zvolen.

## 7. Diskuze

Navržené modely transportu objektů v síti staví na základním modelu P systémů (Păun, 1998), které představují distribuovaný, nedeterministický a maximálně paralelní výpočetní model. Tyto charakteristiky platí i pro zde navržené modely.

Výhodou tohoto modelu je možnost simulace velkého počtu procesů v rámci jednoho běhu simulace. Vzhledem k tomu, že při studiu transportu se zřídka zabýváme transportem jedné entity, představují P systémy obecně perspektivní výpočetní model. Současně však P systémy neumožňují sledovat evoluci nebo pohyb jednotlivých objektů (kromě vzácné situace, kdy se v systému nachází jediný objekt). Mezi jednotlivými objekty není rozlišováno, z pohledu modelu se jedná o totožné objekty. Vhodným použitím navrženého modelu transportních P systémů je například modelování disperze, kdy předpokládáme velké množství látky v počátečním místě a sledujeme, jak se v průběhu výpočtu mění její množství v dalších částech systému. Transportní P systémy umožňují kromě přemísťování objektu měnit i jeho kvalitu. Změna počtu objektů v systému je možná pouze skrz komunikaci s vnějším prostředím.

Zatímco paralelismus P systémů je jednoznačně výhodnou vlastností, umožňující efektivní modelování velkého počtu operací současně, nedeterminismus naopak představuje problém při modelování reálných dopravních systémů. Dopravní systémy nejsou nedeterministické, zpravidla jsou dány počátky a cíle přepravovaných objektů a zkoumá se detailně jejich cesta. P systémy jsou sice schopny takové chování popsat, není to však vhodné použití modelu – takto upravený model omezuje nedeterminismus na holé minimum. P systémy jsou tedy vhodné pro modelování transportu, který má spíše mechanickou povahu (například unášení látek vodním proudem) a nejsou vhodné pro modelování procesů, které jsou deterministické a obsahují rozhodovací procesy.

Další obecnou výhodou P systémů je diskrétnost modelovaných objektů. Ve studiích, zabývajících se například populační dynamikou, se mnohdy setkáváme s problémem modelování reliktních, málo početných populací (Barbuti et al., 2008; Colomer et al., 2011). Numerické modely, založené na systému diferenciálních rovnic, nepracují s jednotlivci ale celou populaci



vyjadřují pomocí jedné kvantity (Freedman, 1982). P systémy, pracující s jednotlivými objekty, jsou lépe schopny modelovat málo početné populace, kdy kolaps celé populace může záviset na přežívání několika málo jedinců.

Oba zde navržené modely transportu se zabývají především chováním objektů v membránách, reprezentujících místa v síti a nikoli chováním na komunikacích, které jsou zde reprezentovány pomocí synapsí, spojujících membrány. Transport mezi dvěma membránami je popsán pomocí pravidla, které má přiřazenou časovou konstantu. Není možné zachytit ani chování jednotlivých objektů během transportu mezi membránami (zrychlování, zpomalování, předjížděcí manévry), ani rozdílnou délku cesty.

Výše napsané znamená, že P systémy obecně a zde představené modely konkrétně nejsou použitelné pro modelování detailních interakcí vozidel (či jiných přepravovaných objektů). Ačkoli se tato práce zabývá modelováním prostorových jevů, reprezentovat v P systémech vzdálenost je obtížné. Inspirací pro P systémy byla živá buňka, v níž nezáleží na vzdálenosti objektů ale na jejich koncentraci (Păun, 1998). Předpokládá se, že vše, co se nachází v jedné membráně je dostatečně blízké na to, aby spolu mohlo interagovat, zatímco vše, co je v rozdílných membránách, je příliš vzdálené, aby spolu mohlo interagovat. V rámci objektů, nacházejících se v membráně, neexistuje pořadí, pouze koncentrace. Snaha definovat v rámci P systémů vzdálenost je proto v protikladu s myšlenkovým rámcem, z něhož tento model vzešel. Tato práce neměla ambice rozšířit model P systémů o práci se vzdáleností nebo obecně prostorem, pouze v rámci již publikovaných rozšíření našla principy, které umožňují s prostorem nepřímo pracovat.

## 8. Závěr

P systémy představují výpočetní model, inspirovaný strukturou a fungováním živé buňky. Základem P systémů je membránová struktura, v jejíž odděleních jsou obsaženy objekty a pravidla, která jimi manipulují. Výpočet probíhá aplikací pravidel na množinu objektů, přičemž dochází ke změnám kvality i kvantity objektů a k jejich přemísťování v rámci membránové struktury.

Tato práce se zabývá možností využití P systémů pro modelování transportu. Téma transportu v membránových systémech bylo široce zkoumáno, téměř výhradně však z pohledu biochemického. Transport v geografickém prostoru je odlišný zejména proto, že se neděje v hierarchické struktuře oddělení (kterou klasické P systémy představují), ale v rovině (případně ve vyšších rozměrech). Práce navazuje na řadu publikací, které se transportem v P systémech zabývaly a představuje dva modely transportu, jež byly sestaveny na základě inspirace fungováním dopravy osob a zboží.

Transportní P systém se signalizací, kapacitami a událostmi modeluje transport objektů v síti, jejíž uzly tvoří membrány. Membrány mohou obsahovat katalyzátory, simulující kapacitu dopravní stavby a synapse spojující membrány mohou nabývat několika stavů, čímž je reprezentována průjezdnost komunikací. Inovativním prvkem je použití událostí, které mají podobu pravidel, která jsou jednorázově aplikována v daném časovém kroku. Pomocí událostí lze simulovat například přerušení komunikace nebo náhlé zvýšení/snížení počtu přepravovaných objektů v některé z částí systému. Je dokázána Turingovská kompletnost navrženého modelu.

Transportní P systém s nosiči modeluje transport objektů pomocí vozidel. Podobá se modelu P systémů s nosiči, v některých stěžejních bodech se od něj však liší. Základem modelu je opět síť membrán, spojených synapsí. Transport objektů mezi membránami však neprobíhá přímo, ale pomocí mobilních membrán (nosičů), které jsou mezi jednotlivými membránami v síti přemísťovány. Mobilní membrány opět obsahují katalyzátory, reprezentující kapacitu nosičů. Pro transportní P systém se signalizací, kapacitami a událostmi je vytvořen simulátor v jazyce R, který je přílohou této práce a umožňuje návrh libovolného P systému na základě představeného modelu a jeho simulaci. V simulátoru jsou provedeny simulace modelových systémů, na kterých je ukázána versatilita modelu. Je provedena simulace

náhodné procházky, z ní odvozené hry ruinování hráče, simulace šíření invazivního druhu říční sítí, simulace šíření znečištění z bodového zdroje v říční síti a simulace přepravy vozidel v jednoduché síti s křižovatkou a železničním přejezdem.

Ačkoli byly oba modely navrženy za účelem simulace přepravy osob a nákladu, větší uplatnění dle mého názoru najdou při simulaci mechanického transportu; například sedimentace, šíření znečištění apod. Osobní doprava je systémem, který je silně deterministický. Nutnost reprezentovat parametry systému, jakými jsou jízdni řády, trasy vozidel a osob, měnící se dopravní poptávka apod. jde proti nedeterminismu P systémů, a vhodnějším nástrojem modelování mohou být například multiagentní systémy, které umožňují modelovat rozhodovací procesy, jež ze své podstaty v P systémech nemohou být vyjádřeny.

## 9. Použité zdroje

ALHAZOV, A., PAN, L., PĂUN, Gh. Trading polarizations for labels in P systems with active membranes. *Acta Informatica*, 41, 2-3, s. 111-144, December 2004.

AMAN, B., CIOBANU, G. Describing the immune system using enhanced mobile membranes. *Electronic Notes in Theoretical Computer Science*, 194, 3, s. 5 - 18, 2008.

BARBUTI, R., MAGGILO-SCHETTINI, A., MILAZZO, P., TESEI, L. Timed P automata. In CIOBANU, G. (Ed.) *Second International Meeting on Membrane Computing and Biologically Inspired Process Calculi*, s. 53-67, 2008.

BARBUTI, R., MAGGILO-SCHETTINI, A., MILAZZO, P., PARDINI, G., TESEI, L. Spatial p systems. *Natural Computing*, 10, 1, s. 3-16, 2011. ISSN 1572-9796. doi: 10.1007/s11047-010-9187-z.

BESOZZI, D., CAZZANIGA, P., PESCHINI, D., MAURI, G. Seasonal variance in p system models for metapopulations. *Progress in Natural Science*, 17, 4, s. 392-400, 2007. doi: 10.1080/10020070708541015.

BLUMEN, A., KLAFTER, J., ZUMOFEN, G. *Models for Reaction Dynamics in Glasses*, s. 199-265. Springer Netherlands, Dordrecht, 1986.

CAVALIERE, M., ZANDRON, C. Time-driven computations in P systems. In GUTIÉRREZ-NARANJO, M. A., PĂUN, G., RISCOS-NÚÑEZ, A., ROMERO-CAMPERO, F. J. (Ed.) *Fourth Brainstorming Week on Membrane Computing*, Sevilla, January 30 - February 3, 2006. Volume I, s. 133-144. Fénix Editora, 2006.

CODLING, E. A., PLANK, M. J., BENHAMOU, S. Random walk models in biology. *Journal of the Royal Society Interface*, 5, 25, s. 813-834, August 2008. doi: 10.1098/rsif.2008.0014.

COLOMER, M. A., LAVÍN, S., MARCO, I., MARGALIDA, A., PÉREZ-HURTADO, I., PÉREZ-JIMÉNEZ, M. J., SANUY, D., SERRANO, E., VALENCIA-CABRERA, L. Modeling Population Growth of Pyrenean Chamois (*Rupicapra p. pyrenaica*) by Using P-Systems, s. 144-159. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. doi: 10.1007/978-3-642-18123-8\_13.

DVORSKÝ, J., JANOŠKA, Z., VOJÁČEK, L. P systems for traffic flow simulation. In CORTESI, A., CHAKI, N., SAEED, K., WIERZCHON, S. T. (Ed.) *CISIM, 7564 / Lecture Notes in Computer Science*, s. 405-415. Springer, 2012.

EPSTEIN, R. A. (Ed.). *The Theory of Gambling and Statistical Logic* (Second Edition). Boston : Academic Press, second edition edit. ISBN 978-0-12-374940-6.

FELLER, W. An Introduction to Probability Theory and Its Applications. 1. Wiley, January 1968. ISBN 0471257087.

FREEDMAN, H. Deterministic mathematical models in population ecology. Canadian Journal of Statistics, 10, 4, s. 315-315, 1982. ISSN 1708-945X.

FREUND, R., PĂUN, G., PÉREZ-JIMÉNEZ, M. J. Tissue P systems with channel states. Theoretical Computer Science, 2004. In press.

HOOGENDOORN, S. P., BOVY, P. H. State-of-the-art of vehicular traffic flow modelling. Proc. of the Institution of Mechanical Engineers Part I Journal of Systems and Control Engineering, 215, s. 283-303, 2001.

IONESCU, M., SBURLAN, D. On P systems with promoters/inhibitors. Technical Report 01/2004, Dept. of Computer Sciences and Artificial Intelligence, Univ. Of Sevilla, 2004.

KRISHNA, S. N., PĂUN, G. P systems with mobile membranes. Natural Computing, 4, 3, s. 255-274, 2005. ISSN 1572-9796. doi: 10.1007/s11047-005-3771-7.

MADHU, M., KRITHIVASAN, K. P systems with dynamic membrane polarization. Romanian Journal of Information Science and Technology, 4, 1-2, s. 135-154, 2001.

MARTÍN-VIDE, C., PAZOS, J., PĂUN, G., RODRÍGUEZ-PATÓN, A. A New Class of Symbolic Abstract Neural Nets: Tissue P Systems, s. 290-299. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002a. doi: 10.1007/3-540-45655-4\_32. ISBN 978-3-540-45655-1.

MARTÍN-VIDE, C., PĂUN, G., ROZENBERG, G. Membrane systems with carriers. Theoretical Computer Science, 270, 1, s. 779 - 796, 2002b. ISSN 0304-3975. doi: [http://dx.doi.org/10.1016/S0304-3975\(01\)00117-7](http://dx.doi.org/10.1016/S0304-3975(01)00117-7).

OBTULOWICZ, A., PĂUN, G. (In search of) Probabilistic P systems. BioSystems, 70, 2, s. 107-121, July 2003.

PAN, L., ISHDORJ, T.-O. P systems with active membranes and separation rules. Journal of Universal Computer Science, 10, 5, s. 630-649, may 2004.

PĂUN, A., PĂUN, G. The power of communication: P systems with symport/antiport. New Generation Computing, 20, 3, s. 295-305, 2002. ISSN 1882-7055. doi: 10.1007/BF03037362.

PĂUN, G. Computing with membranes. Technical Report 208, Turku Center for Computer Science-TUCS, 1998. ([www.tucs.fi](http://www.tucs.fi)).

PEARSON, K. The Problem of the Random Walk. Nature, 72, 1865, s. 294, 1905.

PÉREZ-JIMÉNEZ, M. J., ROMERO-CAMPERO, F. J. P Systems, a New Computational Modelling Tool for Systems Biology, s. 176-197. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. doi: 10.1007/11880646\_8. ISBN 978-3-540-46236-1.

PESCINI, D., BESOZZI, D., MAURI, G., ZANDRON, C. Dynamical probabilistic P systems. International Journal of Foundations of Computer Science, 17, 1, s. 183-204, February 2006.

STEELE, M. J. Stochastic Calculus and Financial Applications. Springer-Verlag New York, 2001.

SUZUKI, Y., FUJIWARA, Y., TAKABAYASHI, J., TANAKA, H. Artificial life applications of a class of p systems: Abstract rewriting systems on multisets. In CALUDE, C., PAUN, G., ROZENBERG, G., SALOMAA, A. (Ed.) WMP,

SUZUKI, Y., FUJIWARA, Y., TANAKA, H., TAKABAYASHI, J. Artificial life applications of a class of P systems: Abstract rewriting systems on multisets. In CALUDE, C., PAUN, G., ROZENBERG, G., SALOMAA, A. (Ed.) Multiset Processing: Mathematical, Computer Science, and Molecular Computing Points of View, 2235 / Lecture Notes in Computer Science, s. 299-346, Berlin, 2001. Springer-Verlag.

VERLAN, S., BERNARDINI, F., GHEORGHE, M., MARGENSTERN, M. Generalized communicating p systems. Theoretical Computer Science, 404, 1, s. 170 - 184, 2008. ISSN 0304-3975. doi: <http://dx.doi.org/10.1016/j.tcs.2008.04.008>.

## Odborný životopis autora / *Curriculum vitae*

### OSOBNÍ ÚDAJE / *PERSONAL INFORMATION*



---

Jméno / *Name* Zbyněk Janoška  
Bydliště / *Address* Velká nad Veličkou 76, 696 74  
E-mail zbynek.janoska@gmail.com  
Narozen / *Birth* 26. 10. 1986, Hranice na Moravě

### VZDĚLÁNÍ / *EDUCATION*

---

2011–dosud Univerzita Palackého v Olomouci, **doktorské studium**, obor *Geoinformatika a kartografie / PhD study Geoinformatics and Cartography*  
2008–2011 Univerzita Palackého v Olomouci, navazující **Mgr. studium**, obor *Geoinformatika / Master's degree: specialization Geoinformatics*  
2006–2009 Univerzita Palackého v Olomouci, **Bc. studium**, obor *Geografie-Geoinformatika / Bachelor's degree: Geography-Geoinformatics*

### PRAXE / *EXPERIENCE*

---

2015–dosud Česká společnost ornitologická (programátor / programmer) *Vytvoření programu pro evidenci dat o kroužkování ptáků a obsluhu agentury, koordinátor projektu Bezpečné zastávky*  
2014–2015 Elitec software, s.r.o. (programátor / programmer) *Vývoj softwaru pro skladové hospodářství se zaměřením na skladování výbušnin*  
2011–2014 Centrum dopravního výzkumu, v.v.i. (GIS analytik / GIS analyst) *projekty GIROSAF (identifikace shluků dopravních nehod) a OPERA (operativní řízení dopravy)*

### STÁŽE / *MOBILITY*

---

2013 Münster University, Germany

## Seznam vybraných publikací autora / Author's selected publications

### **Publikace v odborném časopise v databázi Web of Science (Jimp )** **Publications in a scientific journal indexed on the Web of Science (Jimp)**

- Bíl, M., Andrášik, R. a **Janoška, Z.**, 2013. Identification of hazardous road locations of traffic accidents by means of kernel density estimation and cluster significance evaluation. *Accident Analysis & Prevention*, 55, pp.265-273.
- Tuček, P. a **Janoška, Z.**, 2013. Fractal dimension as a descriptor of urban growth dynamics. *Neural Network World*, 23(2), p.93.
- Tuček, P., Caha, J., **Janoška, Z.**, Vondráková, A., Samec, P., Bojko, J. a Voženílek, V., 2014. Forest vulnerability zones in the Czech Republic. *Journal of Maps*, 10(1), pp.179-182.
- **Janoška, Z.**, Voženílek, V., a Tuček, P., 2014 Intervalové vymezení průběhu střední linie geografických jevů ze sady polygonů: příklad vymezení regionu Haná z mentálních map. *Geografie*, 119(1), p.91-104.

### **Publikace v konferenčním sborníku** **Papers in conference proceedings**

- **Janoška, Z.** a Dvorský, J., 2012, April. P systems: State of the art with respect to representation of geographical space. In *CEUR Workshop Proceedings-12th Annual Workshop on Databases, Texts, Specifications and Objects, DATESO* (pp. 13-24).
- Dvorský, J., **Janoška, Z.** and Vojáček, L., 2012, September. P systems for traffic flow simulation. In *IFIP International Conference on Computer Information Systems and Industrial Management* (pp. 405-415). Springer Berlin Heidelberg.
- **Janoška, Z.** and Dvorský, J., 2013, duben. P System Based Model of Passenger Flow in Public Transportation Systems: a Case Study of Prague Metro. In *DATESO* (pp. 59-69).
- **Janoška, Z.** and Dvorský, J., 2014. Modelling Queues in Transportation Networks Using P Systems. In *Nostradamus 2014: Prediction, Modeling and Analysis of Complex Systems* (pp. 65-78). Springer International Publishing.



## Annotation

P systems represent distributed, maximally parallel computational model, inspired by structure and functioning of living cell. This thesis explores possibility of using P systems to model transportation. The topic of transportation has been widely explored in membrane computing, mostly from the biological and bio-chemical point of view, however. Transportation in geographical space is different mainly because it is realized in plane (or higher dimensions), instead of in hierarchical structure of compartments, which is common representation in P systems.

This thesis follows up previous publications about transport in P systems and presents two models, inspired in their design by real live transportation of people and goods. Both models are designed primarily to perform simulations and not be used for computation.

*Transport P system with signalling, capacities and events* is designed to model transportation of objects in network of membranes, organized in a oriented graph. The usage is demonstrated on several examples, including both theoretical concepts (random walk in graph and Gambler's ruin) and real live systems (vehicle flow at intersection, transport in a river network). Events are introduced as time-bind rules, which serve to perform singular tasks, that are difficult to represent in common P systems. Turing completeness of the model is proved and software simulator for model in language R is written and published online.

*Transport P system with carriers* is designed to model transportation of object through vehicles (carriers). Innovative feature of this model is usage of mobile membranes as carriers.

Keywords:

P systems, transportation

## Summary

This thesis has aim to explore application of P systems to simulate transport in geographic space. P systems represent novel computational model, inspired by structure and functioning of a living cell (Păun, 1998). P systems are distributed, non-deterministic and maximally parallel models with applications mostly in fields of biology and biochemistry. Many variants of Păun's basic model were introduced, including publications dealing with transport in P systems. These variants take inspiration from transport in cells or tissues, which takes place in fundamentally different scales and is in many respects different from transport in geographical space.

This thesis aims to propose P system based models inspired by vehicular transportation and to show their usage.

***Transport P system with signalling, catalysts and events*** is designed to simulate transportation of vehicles in network, which consists of intersections, white holes (entry points) and black holes (end points). All three types of nodes are represented as membranes, while communication channels between them are represented as synapses (membranes are situated in oriented graph, as in tissue-like P systems (Martín-Vide et al., 2002a). Vehicles are represented as objects, which can have information about next membrane label appended to them. Intersections can be signalled, which is represented through states of synapses (Freund et al., 2004). Only two states (open, closed) are used in all presented use cases, however the number of possible states is not limited. Capacity of intersections is represented by using catalysts, which enable object to pass through synapses. Novel feature of *Transport P systems with signalling, catalysts and events* are events. Event is rule, which is applied only once at given time. Events are useful to represent one-off situations (interruption of communication channel, sudden influx of objects into the system etc.) Events are limited by the same restraints as regular rules (they can be applied only if objects are present in membranes etc.). Rules are of the sort of *communication* rules - objects are transported within membrane structure, but their amount or quality does not change (opposed to *evolution* rules, which change quality and/or quantity of objects) and their application can be conditioned by presence of catalyst and/or synapse in given state. Transport through synapses can vary in time, therefore concepts of timed computations in P systems are applied (Barbuti et al., 2008).

Turing completeness is proven and simulator, written in language R is provided (available at github:

[https://github.com/janoskaz/trans\\_p\\_systems](https://github.com/janoskaz/trans_p_systems)).

Four simulations are presented to demonstrate different fields, in which the model can be used.

**Random walk** is simple stochastic process, which describes a path, consisting of succession of random steps in some mathematical space (Pearson, 1905). Simple random walk is random walk in one direction - i.e. the set of integers, if the step is one. Despite simplicity of concept, random walks find applications in such diverse fields as finance (Steele, 2001), biology (Codling et al., 2008) or physics (Blumen et al., 1986).

*Transport P systems with signalling, catalysts and events* can serve as simulator of random walks (Fig. 1).

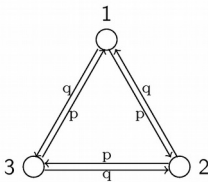


Fig. 1: Membrane system for random walk simulation

Following rule is present on all three membranes:

$$\left[ veh, syn_{\{i,j\}} \right]_i t=1, p \left[ veh \right]_j .$$

The rule transports one instance of object *veh* from membrane *i* to membrane *j* and the operation takes 1 time unit. Each of three membranes *j* contains two rules (to transport object to both neighboring membranes). No other rules are present. In membrane 1 one instance of object *veh* is present, remaining membranes are empty. During computation, object *veh* is being transported through membrane system and from sequence of membranes, which are visited by object, random walk can be inferred. The computation does not stop, therefore number of computational steps must be given in advance.

**Gambler's ruin** is a statistical concept, which can be simplified as *What is the probability, that a gambler with initial wealth will go broke, if he keeps betting until he is either broke or infinitely rich?* Model for Gambler's ruin is random walk and can be simulated using presented model of transportation P systems.

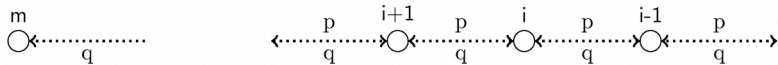


Fig. 2: Membrane systém for simulation of Gambler's ruinování

Membrane system consists of  $M$  membranes, where  $M = A + B$  and  $A$  and  $B$  are initial wealths of two players  $\underline{a}$  and  $\underline{b}$ . Probability of player  $\underline{a}$  winning single round of game is  $p$ , while probability of him losing is  $q$  ( $p + q = 1$ ). In case of fair game,  $p = q = \frac{1}{2}$ . At each round, player wins or loses one unit of wealth and the game continues until one of the players is broke. During computation, object *veh* is transported through membrane system. Label of membrane, in which the object is at given time step, corresponds to current wealth of player. All membranes contain two rules, which transport object to one of neighboring membranes (with given probability). Membranes labeled  $0$  and  $M$  do not contain any rules and if the objects end up in these membranes, the computation ends. For each combination of values  $A, B, p$  a probability of player  $\underline{a}$  being ruined and can be computed (Epstein, 2010). Calculated probabilities correspond to values obtained by repeated simulation of P system.

Moreover, parallelism of P systems can be exploited to run hundreds of simulations during one computation (oposed to classical numerical simulation, where one game at time is simulated). Simulations can be also used to estimate distribution of game duration for given configuration of values  $A, B, p$ . Although estimated duration of game can be calculated (Steele, 2001), empirical distribution can help answer questions like *What is probability, that player will last at least 10 rounds of game?* Etc.

**Dispersion of contamination in river network** is explored using hypothetical river network. One source of contamination is assumed up the stream and three parameters of system are used to represent different scenarios. Probability of contaminant flowing up the stream, down the stream and probability of contaminant staying at current position.

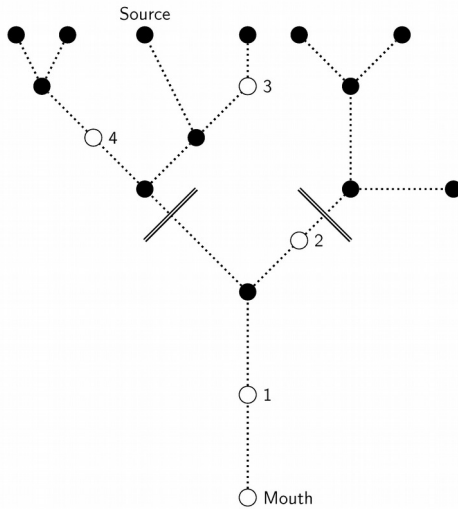


Fig. 3: River network represented as membrane system

Figure 3 shows membrane system used for simulation. Membrane *Mouth* represents mouth of the river, while *Source* is the entry point for contamination. Black membranes represent springs and points of confluence of rivers and white membranes with labels 1 to 4 represent ponds. Different configurations of probabilities are used to explore time, in which the contamination leaves the system (all contaminants are in membrane *Mouth*) and also what are the probabilities of ponds 1 to 4 being contaminated.

Another presented simulations include transport of vehicles through signalled intersection and simulation of dispersion of invasive species in river network. Both examples deal with model systems, however are designed in order to show variability of presented model.

**Transport P systems with carriers** is designed to transport individual objects (passengers) using vehicles – carriers. There are two types of membranes – static membranes (stations) placed into

nodes of a graph and mobile membranes (vehicles). Objects (passengers) enter the system through stations, but can travel between stations only inside vehicles. Originally, simple transport systems like metro were used as inspiration, however capturing public transportation realistically poses many challenges, which are difficult to overcome using the formalism of P systems. The biggest issue is non-determinism of P systems. Real-life transport is deterministic process to a high degree – start and end points of passengers are given in advance, as are routes of vehicles and schedules. Although this thesis discusses, how to incorporate all of these features into presented models, such extension is cumbersome and the resulting variant of presented model lacks features, which make P systems such appealing in first place. For realistic modelling of vehicular traffic flow or passenger flow, other models, such multi-agent systems seem to be better suited. Most attractive features of P systems are non-determinisms and parallelism, therefore phenomena, which inherently have these characteristics are more suitable to be modelled with presented models. Sedimentation, dispersion of contamination of purely theoretical mathematical models like random walks in graphs are only some of topics, where models presented in this thesis can be used.

Mgr. Zbyněk Janoška

**VYUŽITÍ P SYSTÉMŮ PRO MODELOVÁNÍ PROSTOROVÝCH JEVŮ**  
**APPLICATIONS OF P SYSTEMS IN SPATIAL MODELLING**

Určeno pro studenty, partnerská akademická pracoviště a veřejnost.

Výkonný redaktor prof. RNDr. Zdeněk Dvořák, DrSc. et Ph.D.  
Odpovědná redaktorka Mgr. Věra Krischková  
Technická redakce Mgr. Stanislav Popelka

Publikace neprošla redakční jazykovou úpravou.

Vydala a vytiskla Univerzita Palackého v Olomouci  
Křížkovského 8, 771 47 Olomouc  
www.vydavatelstvi.upol.cz  
www.e-shop.upol.cz  
vup@upol.cz

1. vydání  
Olomouc 2015  
Edice GEOINFO-CARTO-THESIS, svazek --.  
ISSN -----  
ISBN -----

Neprodejná publikace