

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA DOPRAVNÍ

SIMULACE OBSLUHY V MENZE

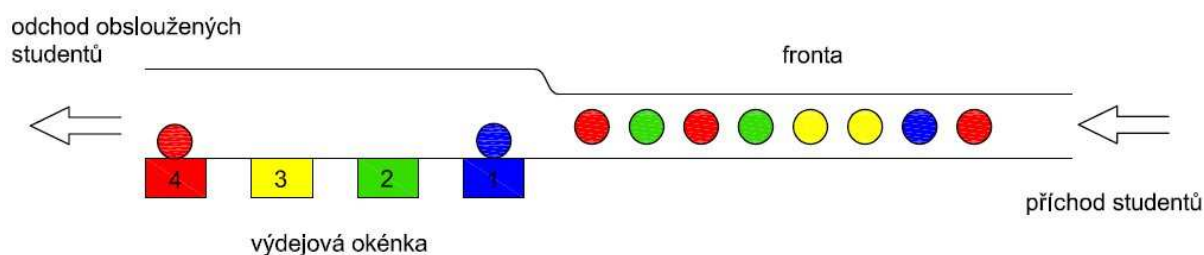
Semestrální práce z předmětu
Teorie hromadné obsluhy

Popis situace v praxi

Pro semestrální práci jsem si vybral obecný zjednodušený případ menzy se čtyřmi výdejovými okénky.

V menze je otevřeno od 11:00 do 14:00. Studenti začínají přicházet do menzy už před 11:00 a tudíž se zde vytvoří fronta už před otevírací dobou. V menze jsou 4 výdejní okénka, tedy studenti si mohou vybírat ze 4 různých jídel. První student stojící ve frontě může přistoupit k výdejovému okénku v případě, že od něj předchozí student už odešel. Studenti stojící ve frontě za ním, ho nemohou předběhnout i kdyby viděli, že u výdejního okénka, ke kterému chtějí jít, je volno. Je to z důvodu stísněného prostoru pro tvorbu fronty a výdej jídel. Když student přijde k výdejovému okénku, zvolí si na automatickém zařízení číslo jídla a přiloží studentskou kartu, pomocí níž je mu z konta odečtena částka, která je účtována za jídlo. Poté si od obsluhy jídlo převezme a odchází obsloužen pryč. Tvorba fronty je naznačena na následujícím obrázku.

Toto je jak již bylo řečeno zjednodušený případ, proto není uvažováno vyzvednutí polévky a případně nápoje. Dále se předpokládá že jídlo nikdy nedojde, tedy že po celou dobu provozu menzy budou dostupná všechna jídla. Každý druh jídla je různě populární, takže se může stát, že většina studentů bude chtít jedno jídlo. To znamená, že jedno výdejní okénko, může být mnohem více vytížené než ostatní.



Formulace problému termíny THO

Příchod studentů

Studenti začínají přicházet do systému už před otevření menzy (simulace začíná už 20 minut před otevřením). A poté samozřejmě přicházejí během celého dne. Byla odhadnuta průměrná hodinová intenzita příchodu studentů do menzy před otevřením: 100 studentů/hod. A průměrná hodinová intenzita příchodu studentů během otvírací doby: 600 studentů/hodinu (v simulaci jsou zkoušeny i hodnoty 500 stud./hod a 700 stud./hod). Protože studenti nejsou nijak ovlivněni a přicházejí z velkého množství studentů bylo zvoleno poissonovské rozdělení příchodu studentů. Jeden krok simulace je roven 1 sekundě. Intenzita příchodu studentů v jednom kroku simulace je tedy $\lambda = 600/3600$ (při uvažování vstupu 600 studentů/hod).

Obsluha studentů

Obsluha studentů u výdejových okének je vždy stejná (skládá se ze stejných úkonů), takže by mohlo být vhodné zvolit pevnou dobu obsluhy. Ovšem vzhledem k tomu, že ne každý student je stejný a ne vždy musí fungovat automatické zařízení na načítání studentské karty napoprvé zvolil jsem si, že obsluha bude z intervalu 8-12 sekund. Nejpravděpodobnější je 10s, méně pak 11 a 9 s a nejméně 8 a 12s. Tedy podle normálního rozdělení se střední hodnotou 10 a rozptylem 1. Doba obsluhy se v simulaci generuje tak, že jsou generovány hodnoty normálního rozdělení se střední hodnotou 10 a rozptylem 1. Pokud padne hodnota do intervalu:

9,5-10,5 ... je brána jako 10

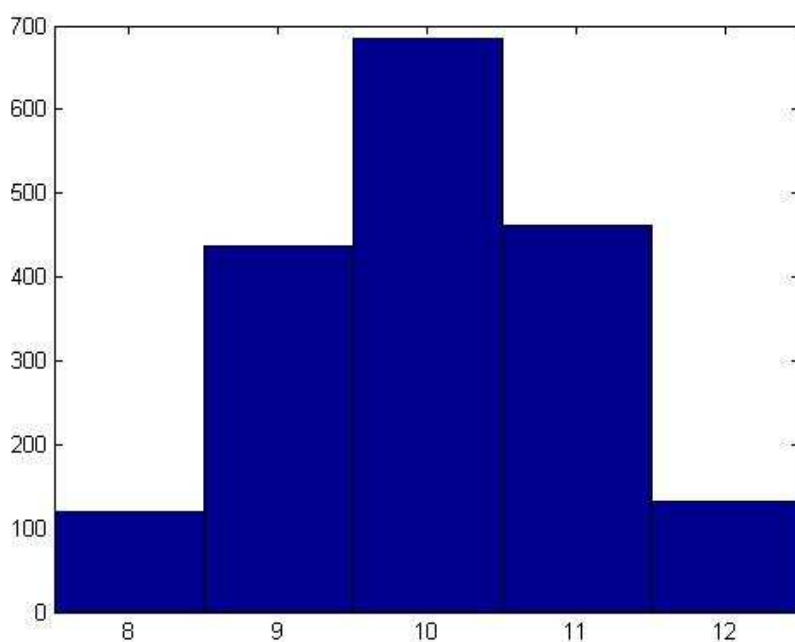
10,5-11,5 ... je brána jako 11

8,5-9,5 ... je brána jako 9

11,5 a více ... je brána jako 12

8,5 a méně ... je brána jako 8

Př.: histogram pro 1838 studentů je na následujícím obrázku.



Obsluhové přepážky

V systému jsou čtyři paralelně uspořádané přepážky. Všechny mohou obsluhovat najednou.

Fronta

Fronta je typu FIFO (first in first out). První student stojící ve frontě smí přistoupit k výdeji jídla až v případě, že student u okýnka, ke kterému chce jít už odešel. Pro frontu není žádný omezující počet, může nabývat libovolného rozměru.

Kendallova klasifikace

Tento případ je možno popsat pomocí Kendallové klasifikace takto:

M/N/4/∞/FIFO

M – Poissonův proces příchodů

N – doba obsluhy podle normálního rozložení

4 - počet paralelně uspořádaných obsluhových přepážek

∞ - neomezená fronta

FIFO – first in first out

Cíle simulace

Cílem simulace je zjistit jak se mění sledované hodnoty systému v závislosti na změně množství přicházejících studentů a změně preference jídel. Preferencí jídel se myslí předem zadané hodnoty, které vyjadřují jak je každé ze čtyř jídel oblíbeno mezi studenty.

Sledované hodnoty jsou:

průměrný počet studentů ve frontě

průměrná doba obsluhy studenta

průměrná doba čekání ve frontě

průměrný počet obsazených výdejových okének

vytíženost výdejových okének; poměr obsluha:volno

grafické znázornění fronty (závislost množství studentů ve frontě na času)

histogram četnosti dob čekání ve frontě

Řešení simulační metodou

Simulace je provedena v programu Matlab 7.9.0 R2009b. Do hlavního skriptu (*menza.m*) je z důvodu zkrácení vloženo několik podskeptů (*obsluhovani.m*, *rozdeleni_jidel.m*, *odbaveni_studentu.m*, *prichod_k_okenkum.m*, *histogram.m*, *graf_fronta.m*).

Za základní krok simulace jsem zvolil hodnotu 1s. Tedy všechny časové údaje v simulaci jsou udávány v sekundách.

V první části skriptu *menza.m*, po definici základních hodnot (např. doba simulace, atd.) se pomocí cyklu vygeneruje množství studentů, kteří do menzy přijdou a uloží se do pole (vektoru) *prichody*. V poli je pro každou sekundu simulace jeden prvek, jehož hodnota udává kolik studentů přišlo. Je možno nastavit pro každou hodinu simulace různou hodnotu λ a tím zvětšovat nebo zmenšovat množství přicházejících studentů. Součtem prvků tohoto pole tedy získáme počet studentů, kteří za den do menzy přijdou. V simulaci je počítáno s tím, že studenti přicházejí k menze už před otevírací dobou. Počítá se s tím, že studenti začínají chodit už 20 minut před otevřením menzy a začíná se tvořit fronta.

Dále se musí určit každému studentovi jídlo, jaké bude chtít. Využívá se k tomu podskept *rozdeleni_jidel.m*. Na vstupu do *menza.m* se zadává hodnota 4-prvkového vektoru *atrakt_jidel*. Na každém prvku je hodnota v procentech (např. 20% = 0,2) jak moc bude to, které jídlo žádané. Tedy např. *atrakt_jidel* = [0.7, 0.1, 0.1, 0.1] znamená že první jídlo je velmi žádané studenty a ostatní tři jsou žádané méně ale sobě jsou si rovny. Hodnoty jsou uloženy do pole *jidlo*, které má stejný počet prvků jako je počet studentů, každý prvek náleží jednomu studentovi. V každém prvku pole je hodnota 1, 2, 3 nebo 4, podle toho, jaké jídlo student chce. Rozdělování hodnot 1 až 4 mezi studenty probíhá takto: funkcí *rand* je vygenerováno tolik hodnot, kolik je studentů. Poté se v cyklu porovnává do kterého intervalu patří každá vygenerovaná hodnota. Intervaly jsou rozděleny ve stejném poměru jako je poměr hodnot *atrakt_jidel*.

V podskeptu *odbaveni_studentu.m* je na základě normálního rozdělení přiřazena hodnota doby strávené u výdejového okénka každému studentovi. Doby nabývají hodnot od 8 do 12 sekund.

Dále už nastává příprava na samotnou simulaci. Inicializují se proměnné a pole *fronta* se naplní studenty, kteří přišli před otevírací dobou. V simulaci hraje důležitou roli dvojrozměrné pole (matice) *fronta*. Na prvním řádku jsou čísla jídel, které si studenti chtějí dát, ve druhém řádku je doba obsluhy studenta u výdejového okénka, ve třetím je doba strávená ve frontě. Podle toho, jak se vyvíjí fronta, mění se hodnoty *z* a *k*, které ukazují na začátek a konec aktuální fronty. Matice *fronta* je znázorněna na následujícím obrázku.

		z ↓					k ↓			
student č.	1,2,...	55	56	57	58	59	60	61	62	...n
jídlo		1	1	2	2	3	1	2	4	
doba odbavení		10	10	11	8	9	10	11	12	
doba strávená ve frontě		26	36	24	38	28	21	25	29	

V simulaci se v každém kroku zkouší, zda přišel někdo do fronty, pokud ano přiřadí se na konec fronty. Dále se v každém kroku zkouší jestli student, který je na začátku fronty má volné výdejové okénko, pokud ano může být obsloužen a fronta (index z) se posouvá. V každém kroku se zmenšuje doba u výdejového okénka, pokud se dostane na nulu, je student obsloužen a okénko se uvolní pro dalšího studenta.

Na konci skriptu *menza.m* je výpočet výstupních hodnot ze simulace a vykreslení grafu a histogramu. Výstupní hodnoty jsou:

prum_PSVF ... průměrný počet studentů ve frontě

prum_DOS ... průměrná doba obsluhy studenta

prum_DCvF ... průměrná doba čekání ve frontě

prum_POO ... průměrný počet obsazených výdejových okének

vytizenost_oken ... vytíženost výdejových okének; poměr obsluha:volno u jednotlivých výdejových okének

Vstupní hodnoty do skriptu *menza.m* :

lambda ... vektor o 4 prvcích udávající intenzitu vstupu, význam jednotlivých prvků:

λ_1 ... intenzita příchodů před otevřením menzy

λ_2 ... intenzita příchodů v době 11:00-12:00

λ_3 ... intenzita příchodů v době 12:00-13:00

λ_4 ... intenzita příchodů v době 13:00-14:00

krok ... délka jednoho kroku simulace – 1s

atrakt_jidel ... vektor o 4 prvcích udávající atraktivitu jednotlivých jídel, např. [0.4,0.3,0.2,0.1] znamená že 40% studentů bude chtít jídlo č.1, 30% studentů jídlo č.2 atd.

Výsledky

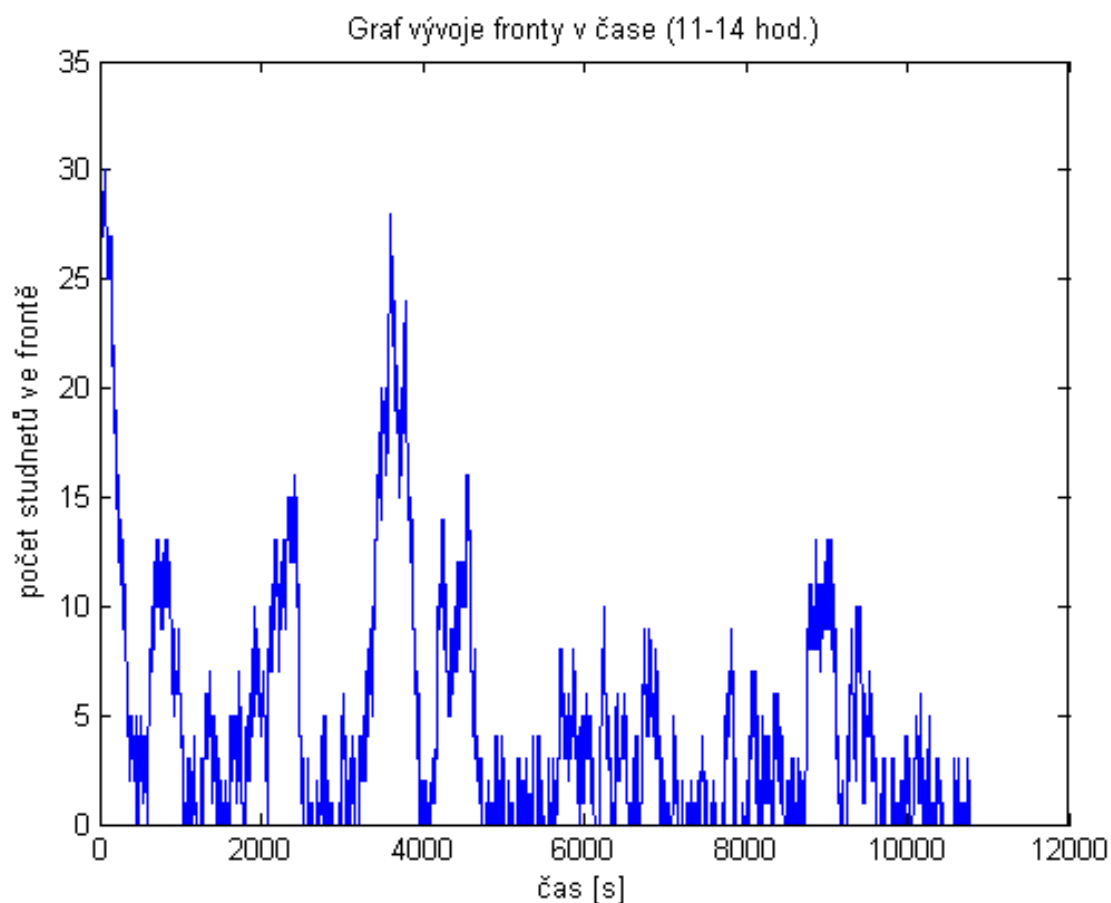
Simulace 1

Vstupní hodnoty:

$\lambda_1=100/3600 \dots 100$ studentů/hod

$\lambda_1=\lambda_2=\lambda_3=600/3600 \dots 600$ studentů/hod

atrakt_jidel = [0.4, 0.3, 0.2, 0.1]



prum_PSVF = 4.558611111111111

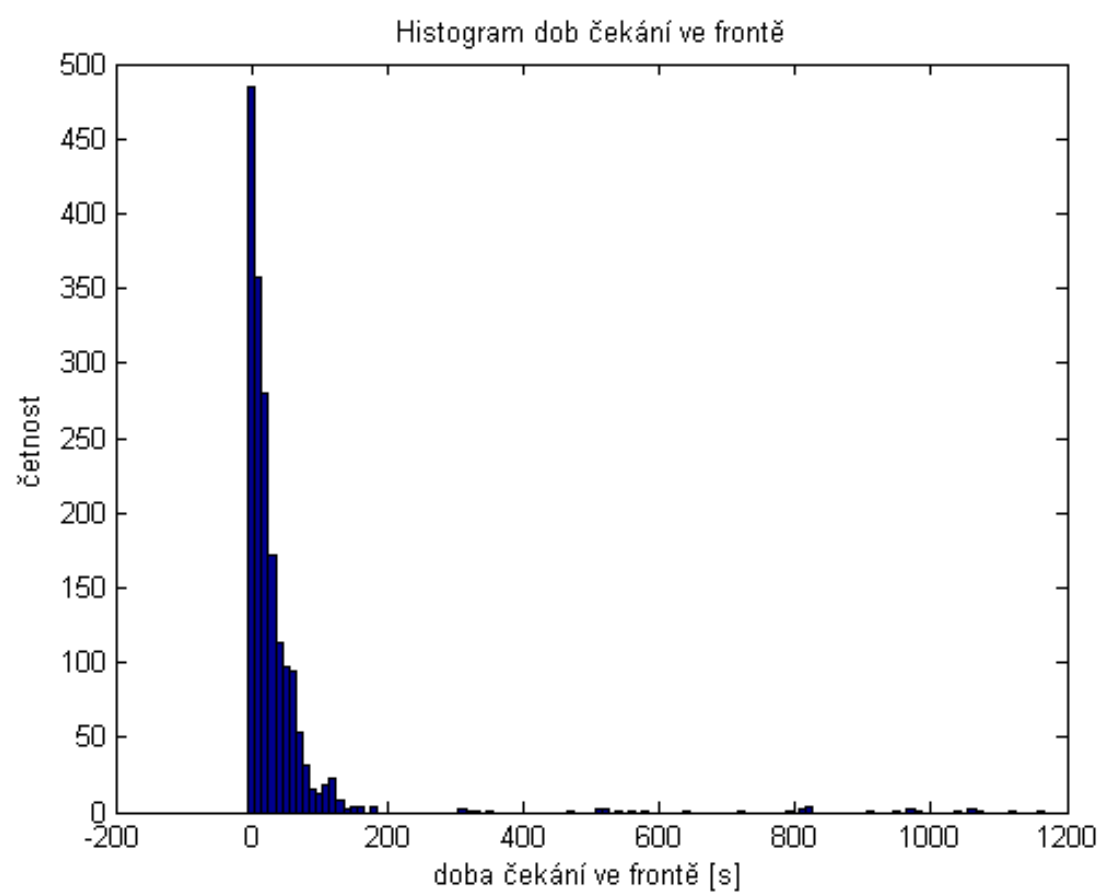
prum_DOS = 10.043815862451471

prum_DCvF = 38.703272323904606

prum_POO = 1.676759259259259

vytizenost_oken =

2.269754768392370 0.912858660998937 0.523702031602709 0.192842942345924



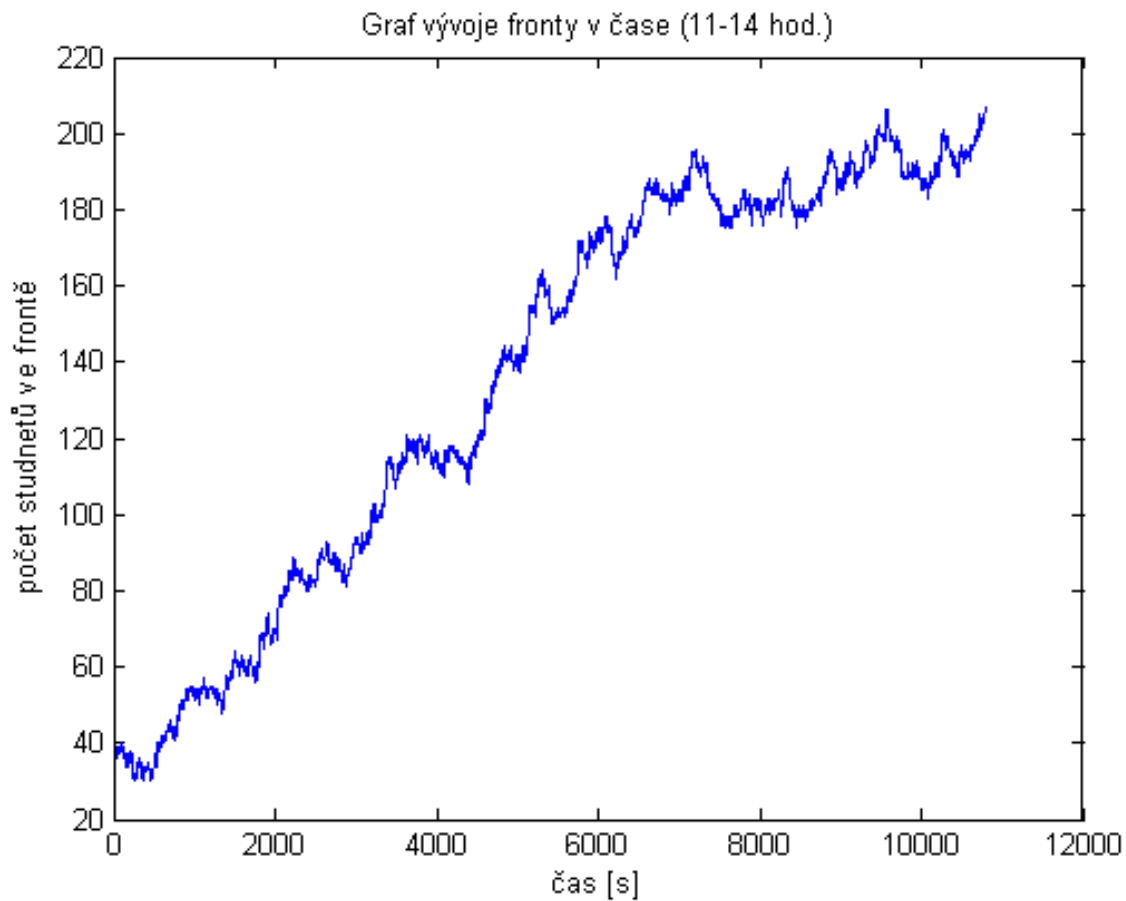
Simulace 2

Vstupní hodnoty:

$\lambda_1=100/3600 \dots 100$ studentů/hod

$\lambda_1=\lambda_2=\lambda_3=600/3600 \dots 600$ studentů/hod

atrakt_jidel = [0.6, 0.2, 0.1, 0.1]



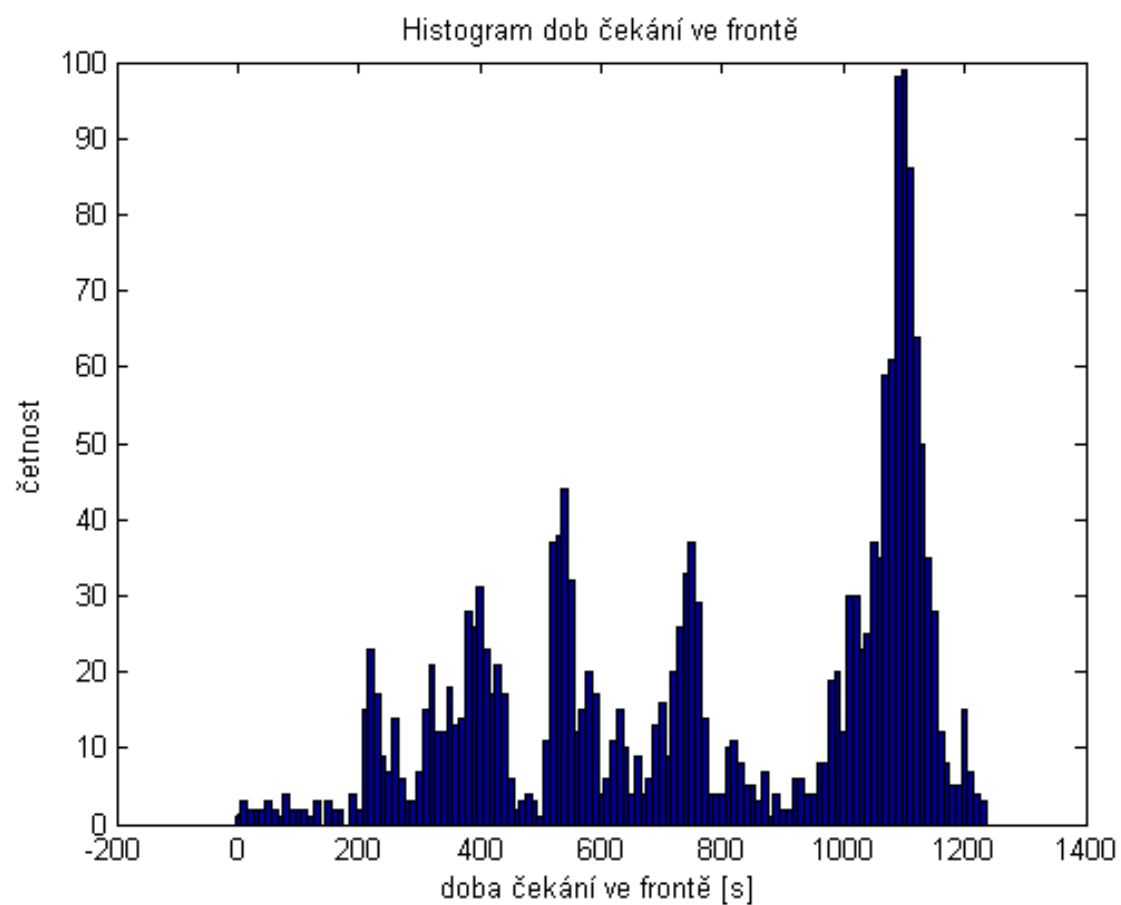
prum_DCvF = 7.851072555205047e+002

prum_DOS = 10.021556256572030

prum_POO = 1.5722222222222222

prum_PSVF = 1.358882407407407e+002

vytizenost_oken = 18.816513761467888 0.470988831381095 0.167315175097276
0.189296333002973



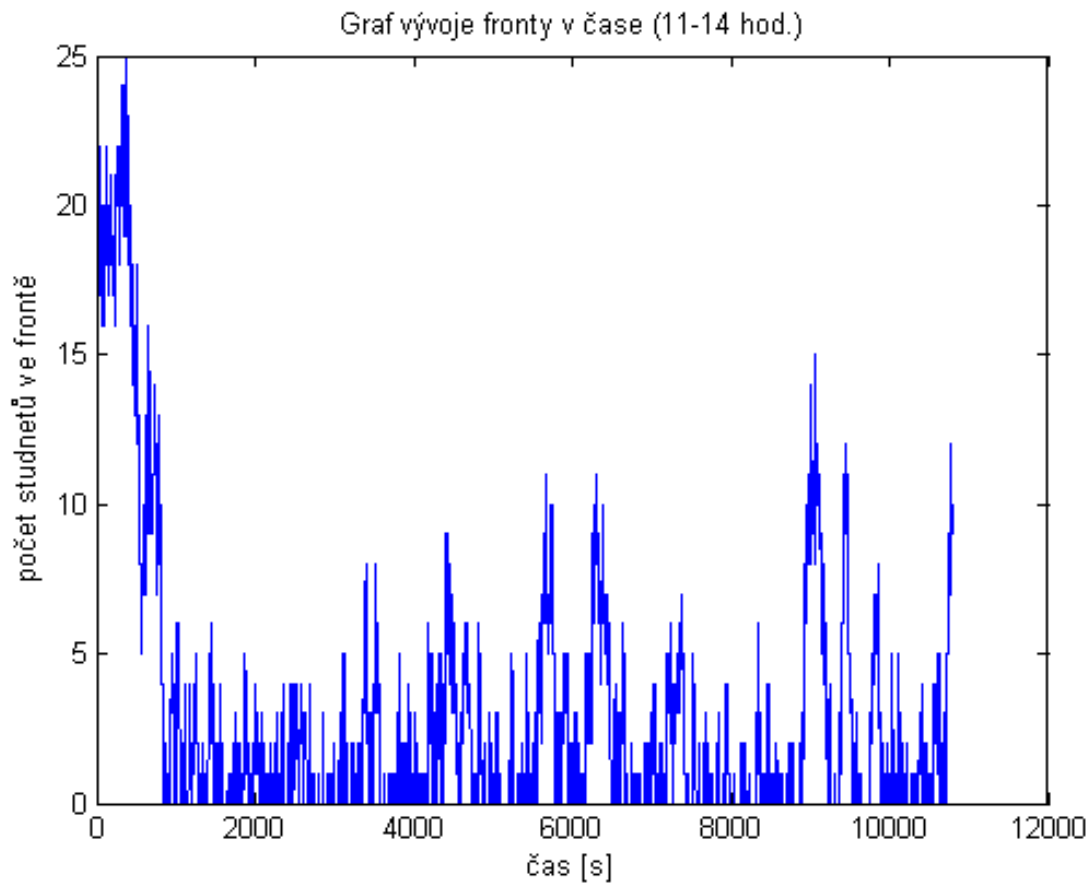
Simulace 3

Vstupní hodnoty:

$\lambda_1=100/3600 \dots 100$ studentů/hod

$\lambda_1=\lambda_2=\lambda_3=600/3600 \dots 600$ studentů/hod

atrakt_jidel = [0.25, 0.25, 0.25, 0.25]



prum_PSVF = 2.904814814814815

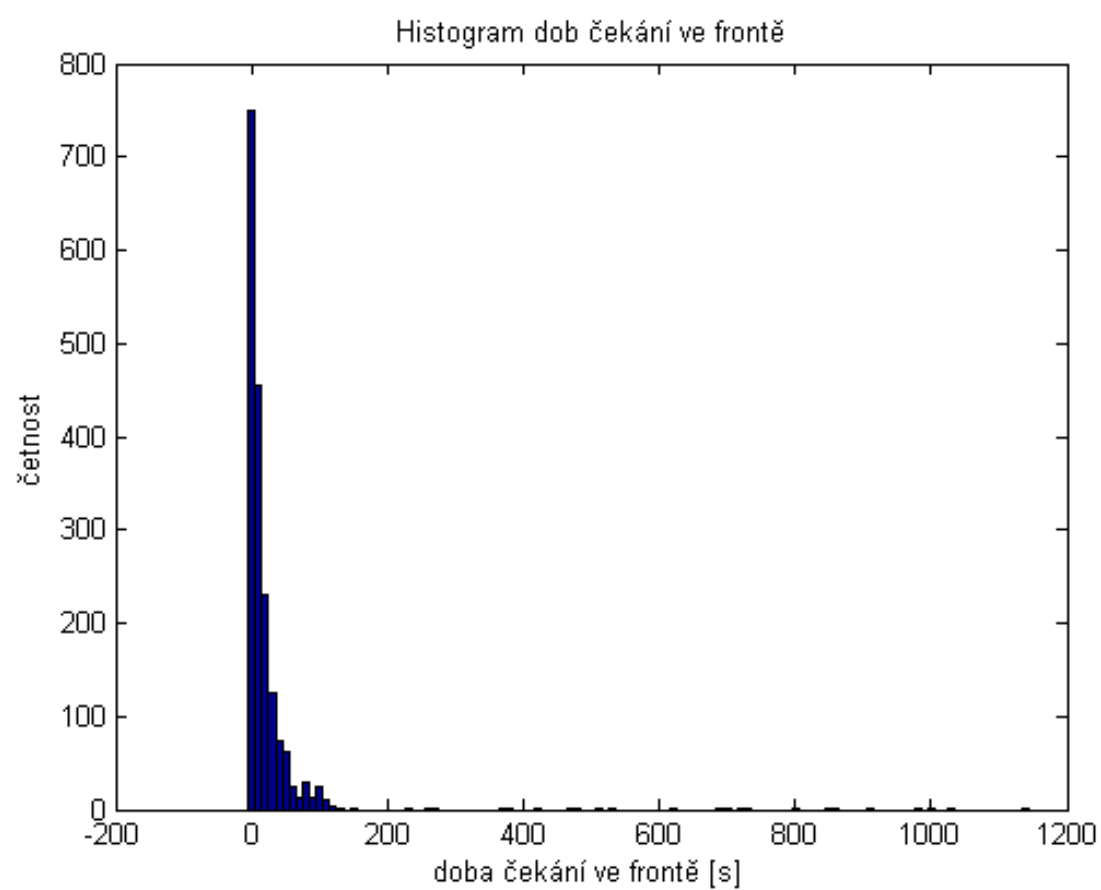
prum_DOS = 10.018448182311449

prum_DCVF = 24.788388497015735

prum_POO = 1.7011111111111111

vytizenost_oken =

0.786009591532992 0.726066805178200 0.766148814390842 0.685130285535965



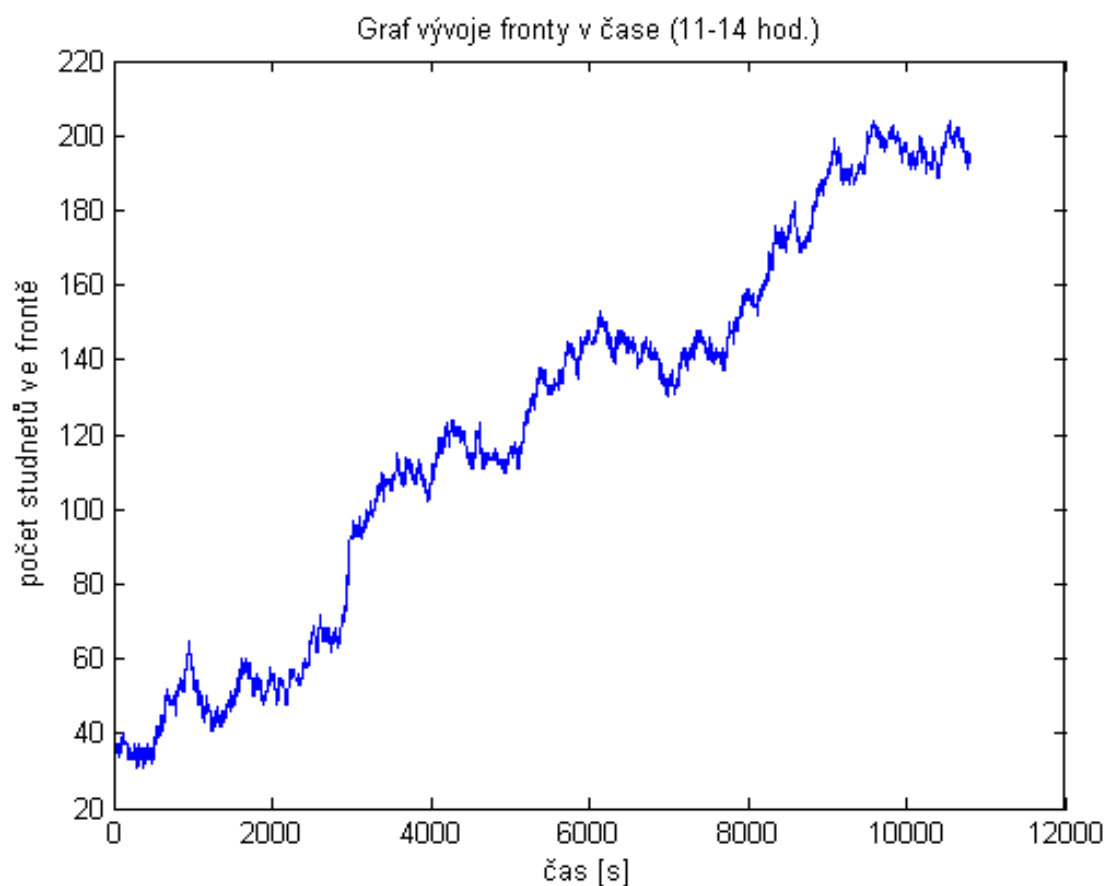
Simulace 4

Vstupní hodnoty:

$\lambda_1=100/3600 \dots 100$ studentů/hod

$\lambda_1=\lambda_2=\lambda_3=700/3600 \dots 700$ studentů/hod

atrakt_jidel = [0.4, 0.3, 0.2, 0.1]



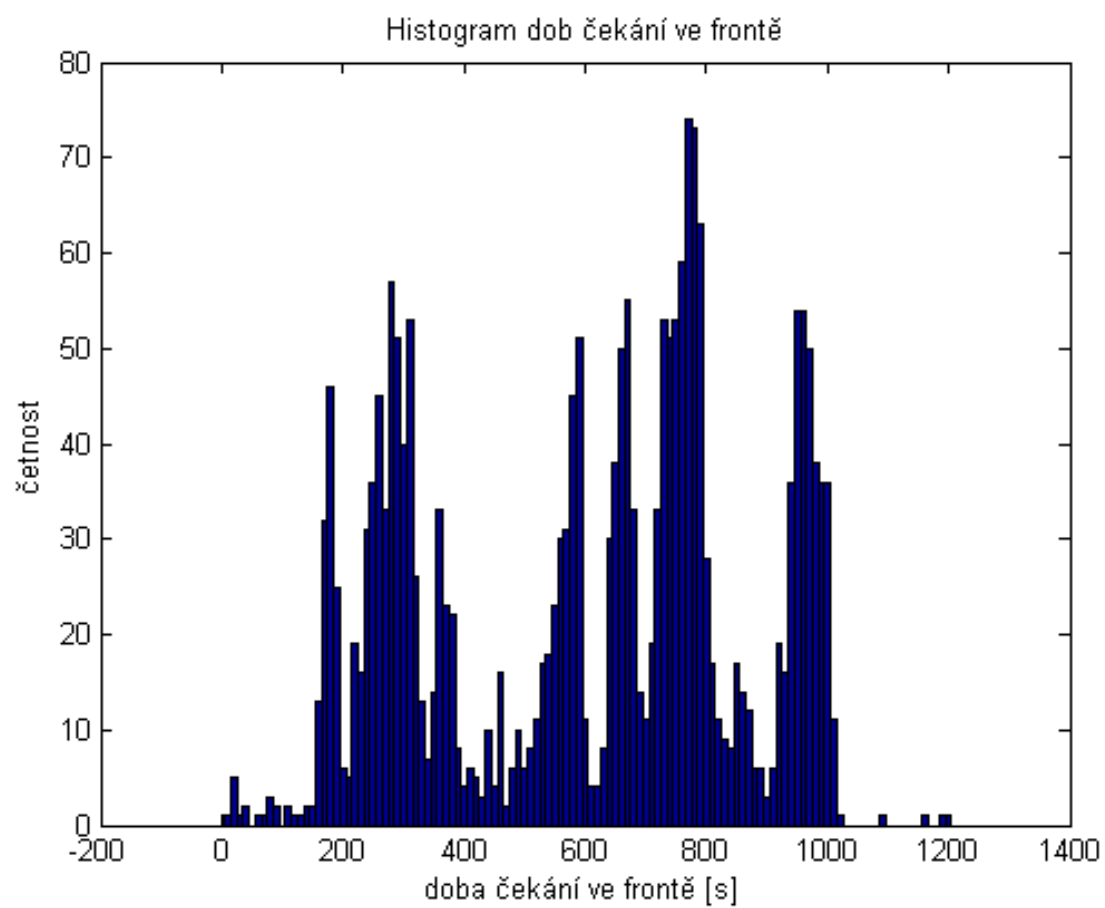
prum_PSVF = 1.2198861111111111e+002

prum_DOS = 10.002712477396022

prum_DCVF = 6.043132911392405e+002

prum_POO = 1.869537037037037

vytizenost_oken = 3.186046511627907 1.240663900414938 0.619190404797601
0.208188835440206



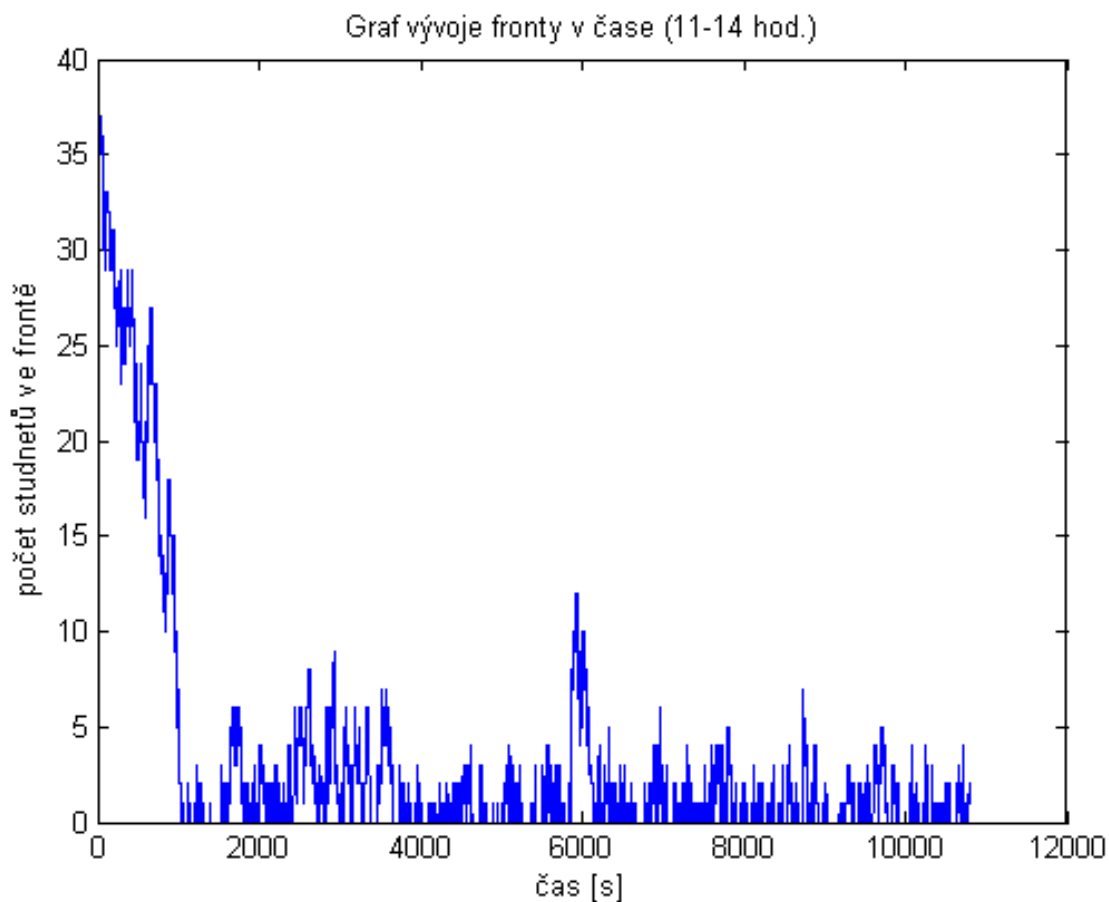
Simulace 5

Vstupní hodnoty:

$\lambda_1=100/3600 \dots 100$ studentů/hod

$\lambda_1=\lambda_2=\lambda_3=500/3600 \dots 500$ studentů/hod

atrakt_jidel = [0.4, 0.3, 0.2, 0.1]



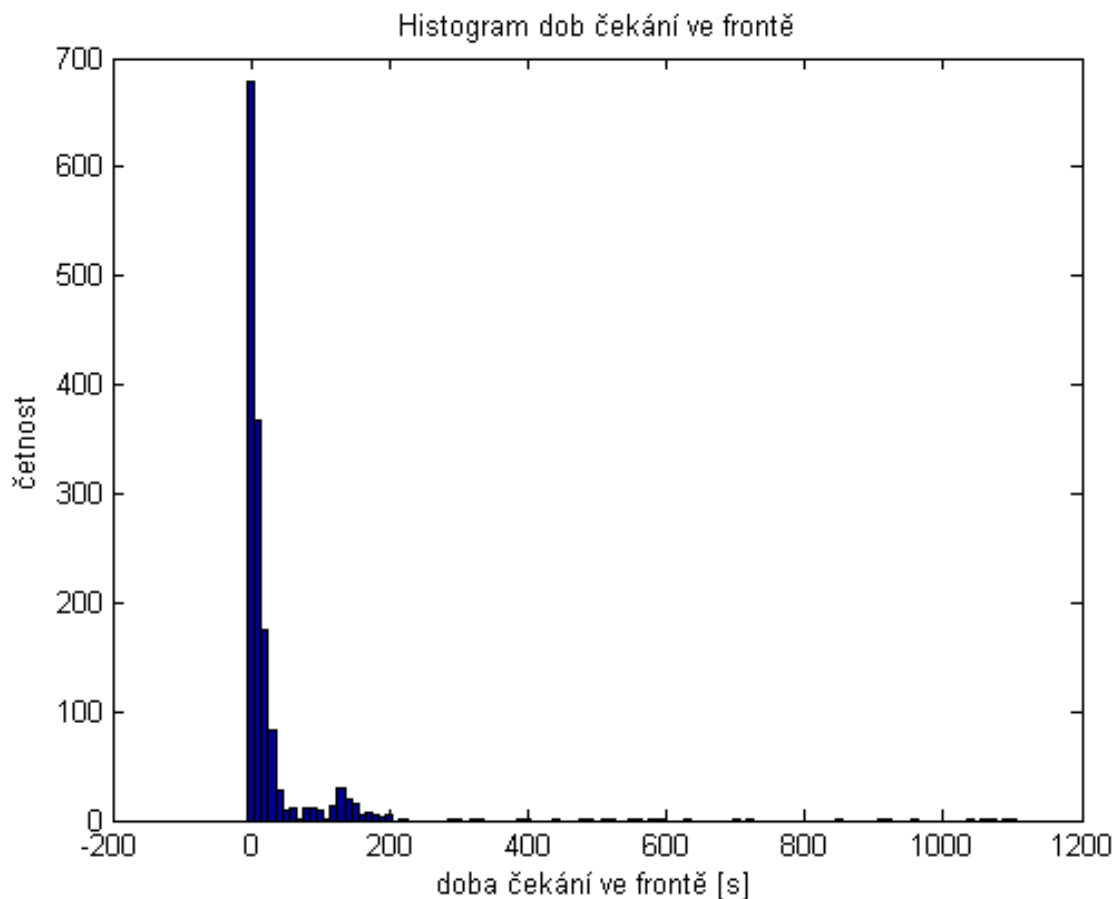
prum_PSVF = 3.252685185185185

prum_DOS = 10.010443864229766

prum_DCVF = 36.119451697127936

prum_POO = 1.4180555555555556

vytizenost_oken = 1.290562036055143 0.808742254228772 0.364842663970681
0.163041137195779



Statistické zpracování – regresní analýza

Pro 2 a 4 simulaci byla provedena lineární regrese. Tedy testujeme, zda počet studentů ve frontě závisí na čase. Z výsledků (uvedených níže) je patrné, že obě závislosti lze proložit přímkou. Korelační koeficienty obou závislostí se přibližují hodnotě 1, což značí silnou vazbu mezi časem a počtem studentů ve frontě. Rovnice pro regresní přímky je $y=b_1*x+b_0$. Hodnota x odpovídá času a vel_fronta je počet studentů ve frontě.

Simulace 2

```
>> [b1,b0,r]=reg_desc(x,vel_fronta)
```

```
b1 = 0.016782881264333
```

```
b0 = 45.252290472711820
```

```
r = 0.960544314673221
```

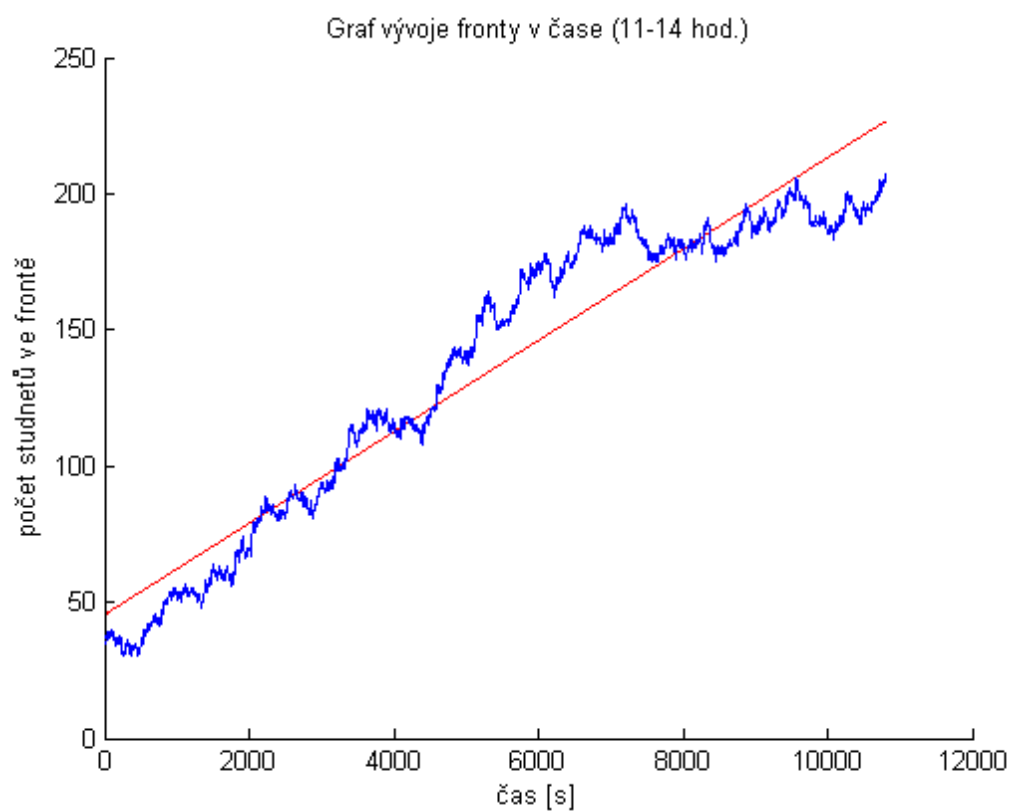
Simulace 4

```
>> [b1,b0,r]=reg_desc(x,vel_fronta)
```

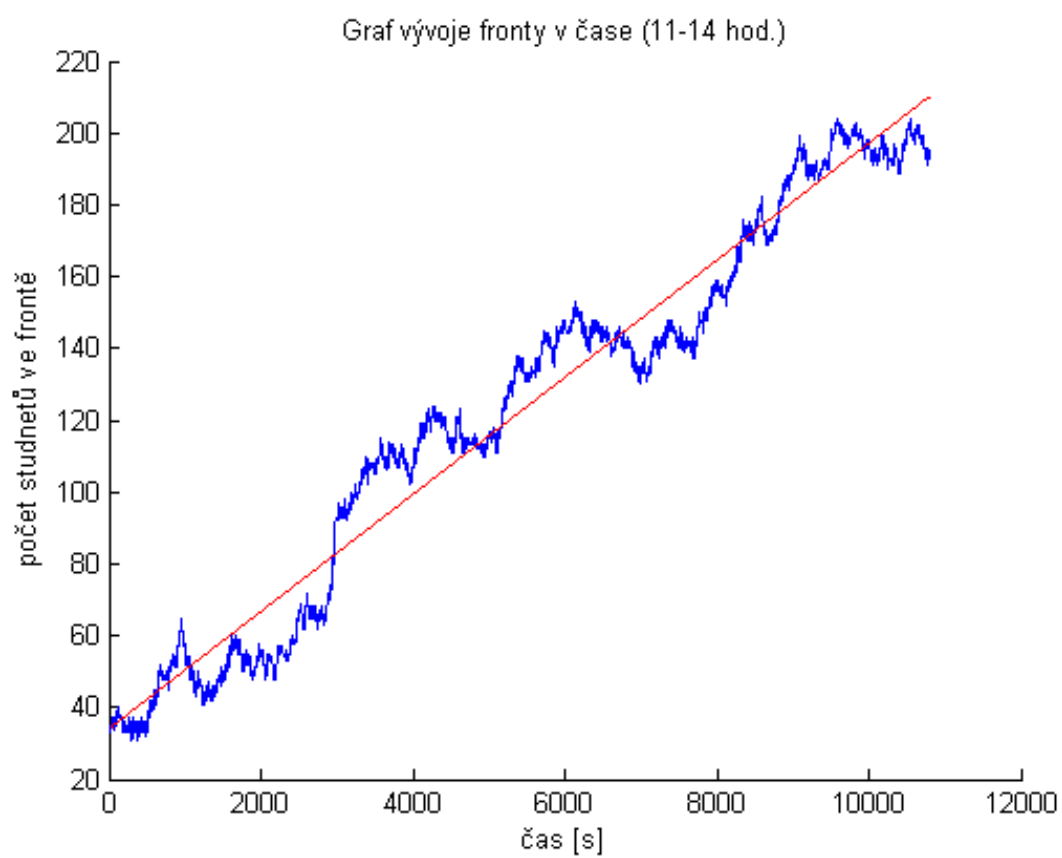
```
b1 = 0.016303202419342
```

```
b0 = 33.943166445452775
```

```
r = 0.980045804141089
```

Simulace 2



Simulace 4

Závěr

Ze simulací je patrné, že pokud do systému přichází 600 studentů/hod. A pokud není jedno z jídel výrazněji studenty preferováno, vyvíjí se fronta vcelku „přívětivě“ a doba čekání ve frontě se pohybuje v únosných mezích. Pokud je však jedno jídlo výrazně preferováno nebo se zvýší počet přicházejících studentů může se stát, že systém nebude stíhat obsluhovat studenty a fronta se během dne bude neustále zvyšovat. V případě snížení počtu přicházejících studentů samozřejmě klesnou čekací doby ve frontě a je vidět, že se fronta během dne téměř nedostává přes hodnotu 10 čekajících studentů.

Výrazné zvýšení výkonu systému by mohlo nastat, pokud by se mohly tvořit čtyři fronty, ke každé přepážce jedna. Tak nemohly nastat situace, že studenti čekající na čele fronty blokuji studenty za nimi, kteří by mohli být obslouženi.

Zdroje:

<http://www.fd.cvut.cz/departament/k611/PEDAGOG/K611THO.html>

Nagy, I., Kratochvílová, J.:Matematická statistika, texty k přednáškám

Matlab: Product help