

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## BAKALÁŘSKÁ PRÁCE



Štěpán Henek

### **Simulace Mraveniště**

Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: RNDr. Ondřej Bojar

Studijní program: Informatika

Studijní obor: Programování

2008

Chtěl bych tímto mnohokrát poděkovat vedoucímu své práce RNDr. Ondřeji Bojarovi, za jeho rady, náměty, připomínky a vůbec všechny čas, který mi věnoval.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 3. srpna 2008

Štěpán Henek

# Obsah

<b>1</b>	<b>Úvod</b>	<b>7</b>
1.1	Napodobované chování . . . . .	7
1.2	Struktura textu . . . . .	9
<b>2</b>	<b>Popis modelu</b>	<b>10</b>
2.1	Prostředí . . . . .	10
2.2	Jídlo . . . . .	10
2.2.1	Kalendář jídla . . . . .	11
2.3	Překážky . . . . .	11
2.3.1	Kalendář překážek . . . . .	11
2.4	Mraveniště . . . . .	12
2.5	Stavební materiál . . . . .	12
2.6	Feromony . . . . .	12
2.7	Mravenec . . . . .	13
2.7.1	Životní cyklus mravence . . . . .	13
2.7.2	Hodnocení . . . . .	13
2.7.3	Genetická výbava . . . . .	14
2.7.4	Kalendář akcí . . . . .	14
2.7.5	Stavy a tah mravence . . . . .	15
2.7.6	Křížení . . . . .	19
2.7.7	Pohyb . . . . .	19
2.8	Monitorování . . . . .	20
2.8.1	Počítadla . . . . .	20
2.8.2	Cesty a Skupiny cest . . . . .	21
2.8.3	Zajímavé proměnné . . . . .	21
2.8.4	Shrnutí . . . . .	21
2.9	Průběh jednoho kola . . . . .	22
2.10	Poznámky k implementaci . . . . .	22

2.10.1	Verze Javy . . . . .	23
2.10.2	Popis souborů . . . . .	23
2.10.3	Rozdíly mezi GUI a konzolí . . . . .	23
2.10.4	Užité datové struktury . . . . .	24
<b>3</b>	<b>Uživatelská Dokumentace</b>	<b>26</b>
3.1	Ovládání přes konzoli . . . . .	26
3.2	Ovládání přes GUI . . . . .	32
<b>4</b>	<b>Porovnání se skutečnými mravenci</b>	<b>40</b>
4.1	Experiment se skutečnými mravenci . . . . .	40
4.1.1	Popis prostředí . . . . .	40
4.1.2	Popis experimentů . . . . .	40
4.1.3	Výsledky pokusů . . . . .	42
4.2	Simulace . . . . .	44
4.2.1	Popis prostředí . . . . .	44
4.2.2	Popis simulace . . . . .	44
4.2.3	Výsledky . . . . .	49
4.3	Srovnání . . . . .	49
4.3.1	Obyčejná . . . . .	49
4.3.2	Zablokování . . . . .	50
4.3.3	Zprůchodnění . . . . .	50
<b>5</b>	<b>Závěr</b>	<b>51</b>
5.1	Diskuze . . . . .	51
5.2	Závěr . . . . .	52
<b>6</b>	<b>Přílohy</b>	<b>53</b>
6.1	CD . . . . .	53
	<b>Literatura</b>	<b>54</b>

# Seznam obrázků

1.1	Ant Colony Optimization . . . . .	8
2.1	Stavy . . . . .	15
4.1	Prostředí experimentu . . . . .	41
4.2	Výsledky reálných mravenců . . . . .	43
4.3	Moje prostředí . . . . .	44
4.4	Výsledky A . . . . .	47
4.5	Výsledky B . . . . .	48

Název práce: Simulace mraveniště  
Autor: Štěpán Henek  
Katedra (Ústav): Ústav formální a aplikované lingvistiky  
Vedoucí bakalářské práce: RNDr. Ondřej Bojar  
e-mail vedoucího: [ondrej.bojar@mff.cuni.cz](mailto:ondrej.bojar@mff.cuni.cz)

Abstrakt: Cílem práce je navrhnout a implementovat simulační prostředí, které bude napodobovat chování mravenců při hledání potravy. Prostředí simulace není po celou dobu statické a může se měnit. Změny mohou být náhodného charakteru, s náhodným prvkem nebo čistě deterministické. Mravenci by na tyto změny měli být schopni zareagovat a přizpůsobit se novým podmínkám. Pohyb mravenců v prostředí je nedeterministický, je ovlivněn stavem bezprostředního okolí a genetickou výbavou každého jedince. Výsledný model je na závěr porovnán s chováním reálných mravenců v umělém prostředí (viz [1]).

Klíčová slova: mravenci, mraveniště, simulace, rojová inteligence

Title: Ant Colony Simulation  
Author: Štěpán Henek  
Department: Institute of Formal and Applied Linguistics  
Supervisor: RNDr. Ondřej Bojar  
Supervisor's e-mail address: [ondrej.bojar@mff.cuni.cz](mailto:ondrej.bojar@mff.cuni.cz)

Abstract: The aim of the paper is to design and implement simulation environment, which will imitate the foraging behaviour of the ants. The environment itself is variable. The changes of environment can be purely random, with random element or purely deterministic. Ants should be able to react to these changes and adapt to new conditions. An ant's movement is non-deterministic influenced by its current surroundings and genetic information. Finally, our model is compared to the behaviour of real ants in artificial settings (see [1]).

Keywords: ant, ant colony, simulation, swarm intelligence

# Kapitola 1

## Úvod

V živočišné říši se vyskytuje mnoho zvířat, jejichž chování by se dalo napodobit značně komplexním algoritmem. Postup, jak reagovat na různé situace, by měl jedince připravit na různá nebezpečí, zajistit pravidelný přísun potravy a v neposlední řadě umožnit úspěšnému jedinci, aby rozšířil svůj genom mezi budoucí generace.

Popsat kompletně něco tak složitého, je ovšem velice obtížné a u živočichů na vyšším stupni vývoje takřka nemožné. V této práci se proto zaměřím na poněkud jednodušší živočichy, mravence. Nebudu se zde navíc snažit popsat kompletní model jejich chování. Půjde mi pouze o krátký výřez z jejich jistě složitě etologie. Konkrétně mi půjde o to, jak napodobit chování mravenců při hledání cesty ke zdrojům potravy. Tento algoritmus bývá někdy označován anglickou zkratkou ACO<sup>1</sup> a lze jej využít například při směřování síťového provozu<sup>2</sup>.

### 1.1 Napodobované chování

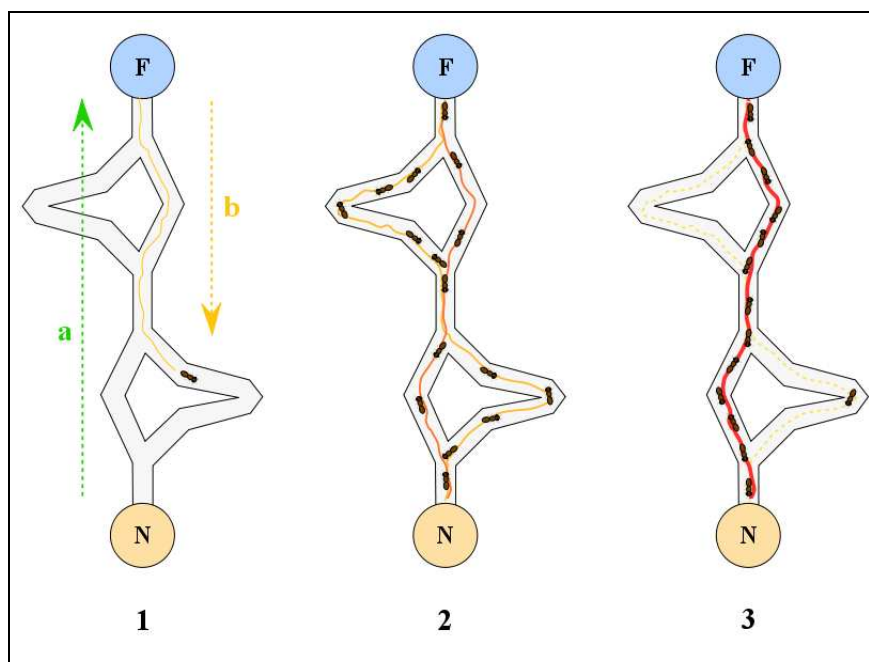
Schéma je poměrně jednoduché. Mravenec se vydává na cestu do neznáma. Snaží se prozkoumat pokud možno co největší území, na kterém by se mohl vyskytovat nějaký zdroj potravy. Když najde jídlo, tak je sebere a začne odnášet zpět do mraveniště. Při návratu vylučuje feromony, které by měly

---

<sup>1</sup>Ant Colony Optimization

<sup>2</sup>[http://en.wikipedia.org/wiki/Ant\\_colony\\_optimization](http://en.wikipedia.org/wiki/Ant_colony_optimization)

přilákat ostatní mravence. Postupným nahromaděním těchto látek se vytváří feromonová stopa. Mravenci se začínají pohybovat po stopě od mraveniště ke zdroji potravy. Když se nějaký mravenec dostane k jídlu jinudy, položí při cestě zpátky základ nové feromonové stopy. Pokud je tato cesta rychlejší, tak ji postupně začne využívat stále více mravenců, protože na udržení určité hladiny feromonů na cestě stačí menší počet mravenců. Koncentrace feromonů se na původní cestě postupně sníží. Původní cesta může zcela zaniknout, případně při větším počtu mravenců může sloužit jako doplněk rychlejší cesty. Takto nalezené cesty nemusí být nejlepším řešením problému. Můžeme je ovšem považovat za jedno lepších, viz Obrázek 1.1.



Obrázek 1.1: Ant Colony Optimization - Obrázek z cs.wikipedia.org



## 1.2 Struktura textu

První kapitola slouží jako stručný úvod a popis toho, o co mi v práci půjde. V druhé kapitole se snažím přiblížit implementovaný model. Třetí kapitola pak slouží jako uživatelská dokumentace. Ve čtvrté je navržený model porovnám s reálnými mravenci. V páté kapitole se nachází diskuze a závěr. Šestá kapitola pouze informuje o přílohách.

# Kapitola 2

## Popis modelu

### 2.1 Prostředí

Okolní svět<sup>1</sup> je reprezentován polem o rozměrech 200x100. Obsahuje právě jeden vchod do mraveniště. Mravenci jej využívají, když potřebují doplnit energii, případně vyložit jídlo nebo stavební materiál.

### 2.2 Jídlo

V prostředí simulace se s určitou pravděpodobností může objevit v příštím tahu nový zdroj jídla. Tato pravděpodobnost se řídí exponenciálním rozdělením (viz [2]).

Jídlo se obvykle objevuje náhodně kdekoli na mapě v náhodném množství (shora i zdola omezeném), případně lze zvolit několik oblastí výskytu. Každá z těchto oblastí má navíc svoji důležitost. V oblastech s větší důležitostí se jídlo objeví s větší pravděpodobností než v těch s menší. Oblasti jsou reprezentovány jako čtverce vymezené dvěma body. Jídlo se objeví v libovolném místě z vybrané oblasti. Pokud se na daném políčku již jídlo vyskytuje, tak se jeho kvantita zvýší. Algoritmus se nesnaží hledat pole bez jídla. Jídlo se může objevit i na nepřístupných polích.

Jídlo je pro mravence životně důležitá surovina. Pokud nebudou mít jídla dostatek, tak se jejich počet začne redukovat. Aby všichni mravenci nevymřeli

---

<sup>1</sup>Budeme jej také označovat jako mapa.

bude zadán minimální počet mravenců. Těmto mravencům se bude během simulace postupně generovat v mraveništi jídlo. Pokud je jídla v mraveništi dostatečné množství dle předem zvoleného parametru, tak se mravenci rozmnoží (viz 2.7.6).

### 2.2.1 Kalendář jídla

Simulace umožňuje za běhu přidávání a mazání oblastí, kde se objevuje jídlo. O tuto vymoženost se stará kalendář jídla. Obsahuje dva typy záznamů, jeden oblast přidá a druhý ji odstraní. Přítomnost záznamu o odebrání není povinná. Pokud neexistuje, tak daná oblast nezmizí.

## 2.3 Překážky

V okolním světě se vyskytují překážky dvojího typu :

- Políčka, která mravenec zpomalí.<sup>2</sup>
- Políčka, které jsou nepřístupná.

Tato políčka se navíc mohou během simulace objevit kdekoliv na mapě podle předem daného kalendáře překážek.

### 2.3.1 Kalendář překážek

Přidávání a mazání překážek během simulace je možné spravovat pomocí kalendáře. Na rozdíl od kalendáře jídla (viz 2.2.1) zde chybí záznam o odebrání a je navíc zadán typ průchodnosti. Tato průchodnost je pak v daném čase aplikována na celou zvolenou oblast. Pokud se oblast stane neprůchodnou, tak jsou všichni mravenci z dané oblasti přesunuti zpět do mraveniště.

---

<sup>2</sup>Průchod trvá 2, případně 3 kola.

## 2.4 Mraveniště

Mraveniště je vlastně základna pro mravence. Mravenci tam nosí jídlo a stavební materiál. Po určité době se mravenec unaví (viz 2.7.3). Musí pak navštívit mraveniště a najíst se. Návštěva mraveniště může mravence zdržet na jedno nebo více kol, dle náhodné hodnoty, jejíž rozsah je určen parametrem.

Mraveniště má od začátku omezenou kapacitu (tj. maximální počet mravenců v simulaci). Tato kapacita lze rozšířit tím, že mravenci nanosí potřebné množství stavebního materiálu. Když se volná kapacita blíží nule, tak mravenci začnou nosit do mraveniště mimo jídla i stavební materiál.

## 2.5 Stavební materiál

Mravenci potřebují stavební materiál k rozšíření svého mraveniště. Materiál se vyskytuje na mapě náhodně, v předem zadaném množství. Na jednom políčku je maximálně jedna jednotka stavebního materiálu. Samovolně nemizí a nově se objevuje pouze, když mravenec nějaký kus stavebního materiálu sebere. Díky tomu se bude na mapě se vyskytovat vždy konstantní počet políček se stavebním materiálem.

## 2.6 Feromony

Hlavní komunikační nástroj mezi mravenci jsou feromony. Mravenci je postupně vylučují během cesty<sup>3</sup>, aby dali zprávu ostatním. Existují dva typy feromonů – A a B. Feromon A je vylučován při hledání potravy a feromon B při návratu s potravou do mraveniště. V jiných situacích mravenec žádný feromon nevylučuje. Feromony se postupem času z políček vypařují, až se nakonec vytratí úplně. Počet vylučovaných jednotek feromonů, maximální počet jednotek na políčku a míra vypařování jsou nastavitelné.

---

<sup>3</sup>Na políčko, kde stojí a na okolní volná políčka

## 2.7 Mravenec

Mravenec je jediná věc, která se na mapě pohybuje. Má čtyři základní úkoly. Najít jídlo a stavební materiál, přenést jídlo a stavební materiál do mraveniště, pokusit se předčasně neukončit svoji existenci a šířit svůj genom. Výkonává přitom 4 základní akce:

**jdi** Mravenec se přesune z jednoho pole na druhé.

**otoč** Mravenec se otáčí do požadovaného směru.

**seber** Mravenec bere požadovanou surovinu.

**vylož** Mravenec pokládá nesnou surovinu.

### 2.7.1 Životní cyklus mravence

Mravenec potřebuje k životu energii. Tu získává při konzumaci jídla v mraveništi. Pokud energie mravence klesne na nulu mravenec umírá. Pokud není v mraveništi dostatek potravy, tak se mravencům, kteří mají nízké hodnocení (viz 2.7.2), zmrazí příděl potravy. Mravenec nedostane najíst, když je v mraveništi počet jednotek jídla nižší než jeho lepší pořadí v žebříčku nosičů a žebříčku objevitelů.

např. *Je-li mravenec osmý nejlepší nosič a šestý nejlepší objevitel, tak dostane najíst, jen pokud je v mraveništi více než pět jednotek jídla.*

Mravenec má také omezenou dobu života. Po jejím uplynutí umírá a přenechá své místo mladším generacím. Tato doba je zadávána jako parametr simulace, všichni mravenci ji mají stejnou a nemůže se během simulace měnit.

### 2.7.2 Hodnocení

Pro účely křížení a přidělování potravy se mravenec hodnotí podle:

1. Množství jídla, které nanosil do mraveniště.  
(=**nosič**)
2. Počtu polí bez feromonů, kterými prošel.  
(=**objevitel**)

Aby se mohla budoucí generace mravenců lépe prosadit, tak je během křížení u všech mravenců hodnocení sníženo. Z hodnocení dle jídla se dvě dodávky smažou a objevitelské zásluhy jsou zcela vynulovány.

### 2.7.3 Genetická výbava

Genetická výbava u každého jedince je udržována v poli **GEN**. Hodnoty pole **GEN** jsou reálná čísla v rozsahu:

$$\text{GEN}[0], \text{GEN}[14] \in \langle 0, 10 \rangle \quad (2.1)$$

$$\text{GEN}[1], \dots, \text{GEN}[13] \in \langle -5, 5 \rangle \quad (2.2)$$

První položka pole ovlivňuje, kdy se mravenec začne vracet zpět do mraveniště. Je to vlastně přechod ze stavu **HLEDAT** (viz 2.7.5) do stavu **VRA-TIT** a závisí také na současné energii mravence(=em) a parametru simulace zvaném energetická hranice(=eh). Přechod se uskuteční právě když :

$$\text{em} \leq \text{eh} * (1 + \text{GEN}[0] / 10) \quad (2.3)$$

Díky tomu má genetická výbava vliv na to, kdy se mravenec začne vracet. Navíc bylo potřeba zajistit, aby mravenci se nezačali vracet zpět moc pozdě/brzy, což zajišťuje parametr eh. Jinými slovy mravenec se může samovolně začít vracet když:

$$\text{em} \in \langle \text{eh}, 2 * \text{eh} \rangle \quad (2.4)$$

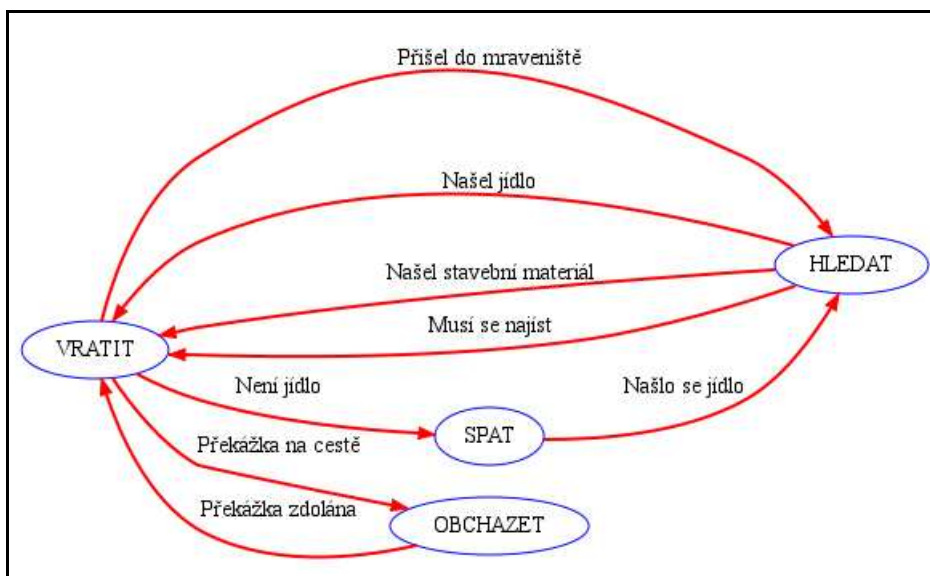
v závislosti na jeho genetické výbavě.

Další položky tohoto pole mají vliv na výběr políčka, kam se mravenec pokusí přesunout při stavu **HLEDAT**.

### 2.7.4 Kalendář akcí

Mravenci jsou uloženi do kalendářové struktury. Každý mravenec je zde asociován s časem jeho příští aktivace. Při aktivaci mravenec vykoná nějakou akci, následně je mu naplánován čas jeho příští aktivace.

## 2.7.5 Stavy a tah mravence



Obrázek 2.1: Stavy mezi kterými může mravenec přecházet

Mravenec se může nacházet v různých stavech. Podle těchto stavů a stavu okolí<sup>4</sup> s jistou pravděpodobností rozhodne co dělat v následujícím tahu. Tato pravděpodobnost je ovlivněna genetickou výbavou každého mravence (viz 2.7.3). Přejchod mezi stavy může nastat při návštěvě mraveniště, při nalezení potravy, při nalezení stavebního materiálu, při nalezení překážky na cestě a samovolně v závislosti na tom, jak je mravenec unavený (tj. kolik mu zbývá energie), viz Obrázek 2.1.

### Stav SPAT

V tomto stavu mravenec pouze čeká v mraveništi, dokud se pro něj nenajde nějaké jídlo. Toto čekání ho stojí 3x méně energie, než obvykle.

Mravenec se nezapisuje do kalendáře událostí na příští tah. Zapíše se na tah, který je vzdálen tři jednotky času od současného. Mravenci se odečítá energie na počátku jeho tahu. Jakmile je na tahu, podívá se, jestli by se pro něj

<sup>4</sup>mravenci vidí jen políčka ve své bezprostřední blízkosti

nenášlo jídlo. Pokud ne, opět se uspí. Když se nají, tak přechází do stavu HLEDAT.

### Stav HLEDAT

V tomto stavu se mravenec snaží najít zdroj jídla, stavební materiál, případně prozkoumat co nejvíc neobjevených polí (pole na nichž se nevyskytují feromony). Pouze v tomto stavu je mravenec schopen sebrat jídlo a stavební materiál. Tato akce může trvat i delší dobu, která závisí na nastavení simulace.

Mravenec může přejít z tohoto stavu pouze do stavu VRATIT. Přechází pokud sebere jídlo nebo stavební materiál, případně pokud jeho energie klesne pod určitou hodnotu (viz 2.7.3).

Na základě přítomnosti feromonů v bezprostředním okolí, své genetické výbavy a svého aktuálního směru si mravenec zvolí políčko, na které se pokusí dostat. Pokusí se natočit do směru, ve kterém je políčko dosažitelné. Pokud se během otáčení pole znepřístupní, je mravenec nucen volit pole znovu. Díky tomu by se dalo říct, že se v tomto stavu se mravenec nachází ve dvou podstavech. V prvním se otáčí a ve druhém jde na jiné políčko. Otáčení může být navíc přerušeno přechodem do stavu VRATIT.

Výběr políček probíhá tak, že se všechna políčka z okolí mravence ohodnotí a seřadí dle hodnocení. Pravděpodobnost, že bude políčko vybráno, závisí na jeho ohodnocení a na pořadí mezi ohodnocenými políčky. Tento způsob výběru políčka nám zajistí, že ne vždy je vybráno pole s největším hodnocením, ale šanci dostanou i políčka s podobně vysokým hodnocením.

Políčka se ohodnotí dle pole GEN (viz 2.7.3):

- a) Pokud se na políčku vyskytuje **Feromon A** v množství mA.

$$h = \text{GEN}[1] * \sin(\text{GEN}[2] * \text{mA}/\text{maxA} + \text{GEN}[3]) \quad (2.5)$$

maxA je maximální množství feromonu A na políčku



b) Pokud se na políčku vyskytuje **Feromon B** v množství  $mB$ .

$$h = \text{GEN}[4] * \sin(\text{GEN}[5] * mB/\text{maxB} + \text{GEN}[6]) \quad (2.6)$$

$\text{maxB}$  je maximální množství feromonu B na políčku

c) Pokud se na políčku vyskytuje **Feromon A i Feromon B**.

$$h = \text{GEN}[7] * \sin(\text{GEN}[8] * mA/\text{maxA} + \text{GEN}[9]) + \\ + \text{GEN}[10] * \sin(\text{GEN}[11] * mB/\text{maxB} + \text{GEN}[12]) \quad (2.7)$$

d) Pokud se na políčku nevyskytuje **žádný Feromon**.

$$h = \text{GEN}[13] \quad (2.8)$$

- Pokud se mravenec musí otáčet, aby se dostal hodnocené políčko, zvyšujeme hodnocení:

$$h := h - \text{GEN}[14] / 5 * \text{počet otočení} \quad (2.9)$$

- K výsledku se přičte náhodná hodnota  $\varepsilon$  reprezentující chybu receptorů. Tato chyba nám pomáhá v situacích na počátku simulace, kdy se na políčku nevyskytují feromony. Mravenec si díky tomu nevybírá pořád ty stejná pole. Hodnota chyby je dostatečně malá, takže nebude mít větší vliv na volbu políčka v pozdějších fázích simulace.

$$\varepsilon \in \langle 0, 0.01 \rangle \\ h := h + \varepsilon \quad (2.10)$$

Díky použití funkce sinus má genetická výbava vliv na:

- určení intervalů hodnot, které mravence přitahují/odpuzují
- nastavení velikosti těchto intervalů

## Stav VRATIT

Tento stav zodpovídá za pokud možno bezproblémový návrat mravence do mraveniště. Díky možnosti volby, zda si bude mravenec pamatovat cestu zpět či nikoliv, bylo nezbytné vytvořit dvě verze algoritmu pro návrat:

**S pamětí:** Mravenec se snaží jít zpět do mraveniště dle cesty, kterou vyšel z mraveniště. Z cesty jsou navíc odstraněny smyčky, takže se nemůže stát, že by mravenec přišel na jedno pole dvakrát. Může se ovšem stát, že se mravenci postaví do cesty překážka. Pokud mu cestu blokuje jiný mravenec, tak se pokusí chvíli počkat. Když čekání nezabere, tak odstraní blokována pole z paměti a snaží se dostat na první volné pole na cestě. Zapamatuje si pole kam má namířeno, vzdálenost od toho pole a přejde do stavu OBCHAZET. V případě, že se nejedná o mravence, tak přechází do stavu OBCHAZET rovnou bez čekání a dále postupuje stejně.

**Bez paměti:** Mravenec se snaží jít zpátky do mraveniště podle klesajících hodnot feromonu A, který mravenec pokládá na cestu během svého výletu. Nejprve se snaží jít na políčka s nižším množstvím feromonu A. Náhodně vybírá na které. Potom se snaží dostat na políčka s vyšším množstvím feromonu A. Pole opět vybírá náhodně. Pokud se v jeho okolí nevyskytuje žádné pole s feromonem A, tak postupuje obdobně jako u stavu HLEDAT. Tento postup má smysl pouze, když nastavíme vysoké hodnoty feromonu A v parametrech simulace a navíc se v simulaci musí vyskytovat menší počet mravenců.

## Stav OBCHAZET

Úkolem tohoto stavu je pokusit se obejít překážku, která se postavila mravenci do cesty. Mravenec k tomuto účelu využívá algoritmus pravé/levé ruky<sup>5</sup>. Jakmile se dostane blíž k poli, kam má namířeno<sup>6</sup>, než byl při přechodu do tohoto stavu pole přechází zpět do stavu VRATIT. Pokud se stane, že překážka z ničeho nic zmizí, tak také přechází do stavu VRATIT. Pokud se stane, že pole, kam má namířeno, je volné, ale nedá se na něj dostat, tak má

---

<sup>5</sup>Jde co nejtěsněji podél překážky buď zleva nebo zprava.

<sup>6</sup>první volné pole na jeho cestě zpět

mravenec prostě smůlu a bude se až do své smrti pokoušet obejít překážku. Jestli bude mravenec obcházet překážku z prava či z leva si mravenec volí náhodně. Po celou dobu jeho návratu však obchází překážku pouze z jedné strany.

## 2.7.6 Křížení

Křížení implementované v modelu neodpovídá množení mravenců ve skutečnosti. Neexistuje zde kastovní systém. Díky tomu mohou genetickou informací šířit všichni jedinci, ne pouze královny a samečci. Náš model představuje určité zjednodušení a navíc umožňuje pozorovat větší dynamiku vývoje, která by jinak byla vidět jen při vývoji celých mravenišť.

Zrození nového mravence probíhá tak, že se vezmou čtyři mravenci, dva nejúspěšnější v nošení a dva v objevování. Polovina nových potomků je výsledkem zkřížení dvou nejlepších nosičů, druhá polovina dvou nejlepších objevitelů (viz 2.7.2).

Díky tomu by se postupem času měly vytvořit dvě výjové větve. Jedna specializovaná na objevování, druhá na nošení. Případně by se mohla vytvořit jedna větev univerzálních mravenců, kteří budou zvládat obojí, ale to je vzhledem k odlišnosti jejich úkolů nepravděpodobné.

Samotné křížení probíhá tak, že se při vytváření genetické výbavy nového jedince vezme polovina pole `GEN` (viz 2.7.3) od prvního mravence a druhá polovina od druhého. Navíc může některá položka pole `GEN` s určitou pravděpodobností zmutovat<sup>7</sup>.

## 2.7.7 Pohyb

Na jednom políčku může být vždy maximálně jeden mravenec. Jedinou výjimkou je políčko s mravenišťem, kde se může vyskytovat libovolný počet mravenců.

---

<sup>7</sup>Dostane novou náhodnou hodnotu v daném rozsahu

Mravenec je v každém okamžiku natočen do jednoho z osmi směrů. Budeme je označovat podle světových stran. Při východu z mraveniště je mu přiřazen náhodný směr.

NW	N	NE
W	<b>M</b>	E
SW	S	SE

Tabulka 2.1: Směry

Mravenec může jít přímo (bez otáčení) na tři políčka - na pole, které leží přímo v jeho směru a na pole, která leží v přilehlých směrech jeho původního směru. Během přesunu po přilehlých směrech nedochází, ke změně původního směru.

např. *Pokud je orientován na SE, může jít i na políčka nacházející se ve směru E a S.*

Pokud se chce mravenec dostat na jiná pole, tak se musí otočit. Jedno otočení vždy zabírá jeden tah až tři tahy v závislosti na průchodnosti políčka, na kterém se nachází. Délka přesunu pak závisí na průchodnosti políčka, na které se mravenec přesouvá. Otočením může mravenec změnit svůj současný směr pouze na jeden z přilehlých směrů.

## 2.8 Monitorování

Monitorování v programu je implementováno, tak aby jeho výsledky byly co nejlépe porovnatelné s výsledky v [1].

### 2.8.1 Počítadla

Počítadlo je skupina dvojic souřadnic (tj. množina  $\{(X_1, Y_1), (X_2, Y_2), \dots\}$ ). Má unikátní pojmenování (právě dva znaky) a slouží k počítání mravenců, kteří prošli mezi dvojicemi zadaných bodů. Mezi každou dvojicí bodů vede spojnice. Když ji mravenec překoná, vejde na políčka na spojnici a zase z

nich odejde, tak se zvýší hodnota počítadla. Každá spojnice si udržuje seznam mravenců, kteří se na ní vyskytovali v minulém kole, a na základě rozdílu seznamu mezi minulým a současným kolem si zjistí počet mravenců, kteří jí prošli. Pokud se spojnice jednoho počítadla někde protínají, tak se může stát, že jeden mravenec bude započítán do jednoho počítadla vícekrát<sup>8</sup>.

## 2.8.2 Cesty a Skupiny cest

Cesta je posloupnost počítadel, kterými mravenec prošel při svém putování. Cesty se dále sdružují do skupin. Každá skupina cest má svoje pojmenování (maximálně čtyři znaky), které nemusí být unikátní. Pokud mravenec došel až ke zdroji potravy a prošel všemi počítadly v takovém pořadí, jaké je u některé z cest, tak se hodnota u skupiny cest, do níž daná cesta patří, zvýší o jedna. Před tímto porovnáním jsou z těchto cest odstraněny smyčky, abychom měli lepší přehled o tom, jakou cestu mravenec vlastně použil. Počet cest se smyčkami je pak uložen ve speciální proměnné. Pokud mravencova cesta není ani v jedné skupině cest, tak se zvyšuje počet u výchozí cesty pojmenované *osta*.

## 2.8.3 Zajímavé proměnné

Mimo monitorování cest a počítadel, nás ještě zajímají další proměnné veličiny simulace. Globální čas simulace je nutností, bez něj se nedá určit, kdy se co odehrává. Dále ještě chceme znát počet mravenců a počet jednotek jídla, abychom měli přehled o populačních výkyvech. Méně důležitou vlastností je kapacita mraveniště, z které lze vyčíst pouze to, jak zhruba byla populace mravenců početná v době jejího největšího rozkvětu. Pro statistické se účely se navíc hodí celkový počet sebraného jídla a již zmiňovaný počet cest ke zdroji potravy se smyčkami.

## 2.8.4 Shrnutí

Veškeré výsledky předchozích odstavců jsou použity pro tvorbu výstupů pro statistiky. Ty se pak vypisují přímo na konzoli, případně se dají zobrazit ve

---

<sup>8</sup>Pokud opustí spojnice v průsečíku.

speciálním okně. Statistiky se nevypisují v každém kole, ale pouze jednou za 300 kol. Případně lze v konzolové verzi nastavit časy, kdy se mají statistiky zobrazit.

## 2.9 Průběh jednoho kola

1. Navyšování počítadel (viz 2.8.1)
2. Kontrola kalendářů jídla a překážek (viz 2.2.1, 2.3.1)
3. Zpracování výstupu statistiky (viz 2.8.4)
4. Generování jídla (viz 2.2)
5. Kontrola množení (viz 2.7.6)
6. Tahy mravenců (viz 2.7.4)
7. Zvýšení času simulace
8. Vypařování feromonů

## 2.10 Poznámky k implementaci

K implementaci jsem použil jazyk Java. Zvolil jsem si jej především kvůli možnostem snadné přenositelnosti na různé platformy. Samotná aplikace jde spustit ze dvou verzích:

- Konzolová verze
- GUI verze

Bližší popis tříd, metod a parametrů se dá vyčíst z vygenerovaného javadocu (viz Příložené CD).

### 2.10.1 Verze Javy

Pro přeložení zdrojových kódů je nutné mít nainstalované JDK verze 1.5 a vyšší<sup>9</sup>.

Na spuštění by mělo stačit JRE verze 1.5 a vyšší<sup>10</sup>.

Pro větší pohodlí je možné použít pro překlad program ant<sup>11</sup>.

### 2.10.2 Popis souborů

<i>mraveniste/gui/Ram.java</i>	okno s menu
<i>mraveniste/gui/Pohled.java</i>	vizualizace simulace
<i>mraveniste/gui/RamStatistika.java</i>	okno se statistikou
<i>mraveniste/gui/Vypocet.java</i>	hlavní smyčka simulace
<i>mraveniste/gui/GUI.java</i>	hlavní třída GUI aplikace
<i>mraveniste/gui/dialog/DJidloVyskyt.java</i>	dialog výskytu jídla
<i>mraveniste/gui/dialog/DPromene.java</i>	dialog nastavení vlastností
<i>mraveniste/gui/dialog/DZakladna.java</i>	dialog nastavení polohy základny
<i>mraveniste/gui/dialog/DJidloKalendar.java</i>	dialog nastavení kalendáře jídla
<i>mraveniste/gui/dialog/DPrekazky.java</i>	dialog nastavení kalendáře překážek
<i>mraveniste/gui/dialog/DPocitadla.java</i>	dialog nastavení počítadel a cest
<i>mraveniste/Main.java</i>	hlavní třída konzolové aplikace
<i>mraveniste/Konstanty.java</i>	interface s konstantami
<i>mraveniste/Data.java</i>	třída pro simulaci

### 2.10.3 Rozdíly mezi GUI a konzolí

V konzolové verzi se hlavní smyčka nachází přímo v hlavní třídě, která také pracuje s aktuální instancí třídy Data. Pokud je aplikace spuštěna s parametrem **-gui**, tak je pouze otevřeno okno GUI, žádný další výpočet pak neprobíhá. GUI pro vlastní simulaci využívá třídu Vypocet, ve které běží smyčka. Vypocet pracuje s aktuální instancí třídy Data a běží ve vlastním vlákně<sup>12</sup>.

Hlavní smyčka programu se v GUI a v konzolové verzi liší pouze ve vykreslování a ve výstupních datech. Základní schéma činnosti z 2.9 zůstává za-

<sup>9</sup><http://java.sun.com/javase/downloads/index.jsp>

<sup>10</sup><http://java.sun.com/javase/downloads/index.jsp>

<sup>11</sup><http://ant.apache.org/>

<sup>12</sup>Bez vlastního vlákna by se přestala vykreslovat okna aplikace.

chováno. Cílem toho rozdělení bylo, aby se v konzolové smyčce neprováděly zbytečné akce.

## 2.10.4 Užité datové struktury

- Kalendáře

Do kalendářů jídla a překážek se během simulace nepřidávají prvky. Jediná akce, která se s nimi provádí, je mazání prvního prvku. Toto mazání neprobíhá příliš často. Můžeme očekávat, že proběhne jen párkrát za simulaci. Díky tomu můžeme pro uchovávání prvků použít téměř libovolnou datovou strukturu, aniž by toto rozhodnutí mělo pozorovatelný vliv na rychlost simulace.

Naproti tomu z kalendáře akcí budeme prvky nejen odebírat, ale i přidávat a to poměrně často, protože se do něj zapisuje každý mravenec pohybující se na mapě. Přistupujeme vždy k prvnímu prvku, který následně řadíme dále do fronty. Požadavkům na tuto operaci asi nejlépe vyhovuje datová struktura typu haldy. V jazyce Java je halda reprezentována třídou `PriorityQueue`<sup>13</sup>.

- Mapa

Mapa je reprezentována jako pole. Jednotlivá políčka obsahují pouze statické položky: informace o množství jídla, o stavebním materiálu, o množství feromonů, ... . Tyto údaje bychom mohli udržovat v nějaké dynamické datové strukturu (`LinkedList` políček, co obsahují jídlo, atd.), abychom se vyhnuli zbytečné redundanci. Nicméně by pak během simulace probíhala spousta alokací, které by zpomalovaly výpočet.

- Počítadla

Počítadlo (viz 2.8.1) prochází všechna pole, která obsahuje, a hledá na nich mravence, které následně ukládá do `ArrayListu`. Dále si pamatuje kteří mravenci se na něm vyskytovali v minulém kole. Na základě porovnání těchto dvou `ArrayListů` je možné určit kolik mravenců prošlo počítadlem. `ArrayList` jsem použil, protože je více optimalizován na sekvenční procházení.

---

<sup>13</sup>viz dokumentace k jazyku Java



Jiná možnost by byla, že by každé políčko na mapě mělo seznam počítadel, která zvyšuje. Mravenec by si zapamatoval seznam počítadel políčka, na kterém se nachází. Díky tomu by při přechodu na nové políčko bylo možné zjistit, jak se seznamy liší, a podle toho navýšit určitá počítadla. Tuto možnost jsem nezvolil především proto, že jsem do programu přidával podporu pro počítadla až později a nechtěl jsem příliš zasahovat do již fungující implementace.

- Mravenec

- Cesta zpět

Každý mravenec si přesně pamatuje cestu zpět do mraveniště. Při přesunu na nové pole si ukládá souřadnice tohoto pole do `ArrayListu`. Navíc je zajištěno, aby se v této struktuře nevyskytovaly smyčky. Při přidání nové souřadnice se případná smyčka smaže. Tato kontrola sice stojí určitou režii, ale redukuje spotřebu paměti, která by zvláště při větším počtu mravenců byla neúnosná.

- Navštívená počítadla

Mravenec si pamatuje všechna počítadla, kterými prošel. Díky tomu můžeme zjistit, jakou cestou (posloupností počítadel) přišel ke zdroji potravy. Prvky se zde mohou opakovat. Z této cesty jsou při zpracování vyjmuty smyčky. Tato operace může tedy mazat i prvky uprostřed seznamu a tudíž je pro náš účel vhodnější `LinkedList`.

Mohli bychom mazat smyčky už při přidávání jednotlivých počítadel, nicméně bychom tuto činnost prováděli i u mravenců, kteří cestu k jídlu nenajdou, což je zbytečné. Paměťově je toto řešení sice náročnější, nicméně na rozdíl od cest zpět tu nebude počítadel tolik. Navíc se zde nevytváří nová struktura, pouze se kopíruje odkaz na počítadlo. Dále můžeme počítat s tím, že počet smyček u navštívených počítadel nebude tak velký jako u cest zpět. Navíc chceme znát počet cest se smyčkami, ne celkový počet smyček. Kdybychom je odstraňovali v každém kole, tak by bylo nezbytné si držet u mravence proměnnou, které by říkala, zda prošel smyčkou.

# Kapitola 3

## Uživatelská Dokumentace

Program se obvykle spouští příkazem:

```
java -jar dist/mraveniste.jar [PARAMETRY]
```

### 3.1 Ovládání přes konzoli

PARAMETRY :

**-bp**

Vypíná paměť mravence. Mravenec si ve výchozím nastavení pamatuje cestu zpět do mraveniště. Pokud se k volání přidá tento parametr, tak se mravenec pokusí najít cestu zpět pomocí feromonových stop.

**-nk**

Zapíná nekonečný svět. Když se mravenec dostane na okraj plochy, tak může přelézt a objevit se na opačné straně.

**-mut** *N*

*N* je mezi 0 a 100.

Nastavuje pravděpodobnost mutace při křížení. Pravděpodobnost je rovna  $N/100$ .

**-pocm**  $N$

$N$  je mezi 1 a 100.

Nastavuje kolik mravenců je vytvořeno na počátku simulace. Zároveň se pro tento počet mravenců bude automaticky generovat potrava. Pokud jsou mravenci načtení ze souboru, tak se mravenci na počátku simulace nevytváří.

**-je**  $N M$

$N$  je mezi 1000 a 100000.  $M$  je mezi 200 a  $N$ .

$N$  určuje kolik získá mravenec tahu, když se nají. Pokud mravenec dojde energie, tak umírá.  $M$  určuje tzv. energetickou hranici. Pokud je energie mravence vyšší než energetická hranice, tak se mravenec nenají. Energetická hranice má také vliv na to, kdy se začne mravenec vracet do mraveniště.

**-jo**  $X1xY1:X2xY2[:D[:OD[:DO]]]$

$X1, X2$  jsou mezi 0 a 199.

$Y1, Y2$  jsou mezi 0 a 99.

$D$  je mezi 0 a  $2^{63} - 1$ .

$OD$  je mezi 0 a  $2^{63} - 1$ .

$DO$  je mezi  $OD$  a  $2^{63} - 1$ .

Přidává na mapu novou oblast, kde se bude objevovat jídlo. Pokud nebude žádná taková oblast vytvořena, jídlo se bude objevovat náhodně. Pokud je víc oblastí a pokud má nějaká oblast vysoké  $D$ , tak se tam bude jídlo objevovat častěji než v ostatních oblastech.  $OD$  a  $DO$  nastavují dobu platnosti dané oblasti.

Tento parametr má smysl použít víckrát.

**-p**  $X1xY1:X2xY2[:CAS[:CENA]]$

$X1, X2$  jsou mezi 0 a 199.

$Y1, Y2$  jsou mezi 0 a 99.

$CAS$  je mezi 0 a  $2^{63} - 1$ .

$CENA$  je mezi 0 a 4.

Přidává/Odebírá z mapy překážky.  $CENA$  pak určuje dobu, za kterou mravenec překoná dané pole. Např. Pokud je  $CENA$  3, tak mravenec bude trvat 3 tahy, než pole přejde. Když cena je rovna 0, tak je pole neprůchodné. Pokud při umístování neprůchodné

překážky najdeme na poli mravence, tak jej přesuneme zpátky na základnu. Políčko se základnou je vždy volné a má nastavenou cenu za průchod na 1.

Tento parametr má smysl použít víckrát.

**-z**  $X1 \times Y1 : X2 \times Y2$ .

$X1, X2$  jsou mezi 0 a 199.

$Y1, Y2$  jsou mezi 0 a 99.

Nastavuje novou polohu základny. Zvolené pole je nastaveno jako volné a jeho cena za průchod se nastaví na 1.

**-jp**  $N M$

$N$  je mezi 0 a 1000.  $M$  je mezi 1 a 1000.

$M$  a  $N$  určují kolik tahů vydrží zdroj jídla, než se vyčerpá. Výsledný počet tahů je  $M +$  náhodné číslo mezi 0 a  $N$ .

**-jobev**  $N$

$N$  je mezi 1 a 1000.

Nastavuje pravděpodobnost ( $N / 1000$ ), že se v příštím tahu na mapě objeví nový zdroj jídla.

**-fA**  $N$

$N$  je mezi 0 a 1000.

Nastavuje kolik jednotek Feromonu A vyloučí mravenec na políčko.

**-fB**  $N$

$N$  je mezi 0 a 1000.

Nastavuje kolik jednotek Feromonu B vyloučí mravenec na políčko.

**-maxfA**  $N$

$N$  je mezi 1000 a 100000.

Nastavuje maximální hodnotu Feromonu A na políčku.

**-maxfB**  $N$

$N$  je mezi 1000 a 100000.

Nastavuje maximální hodnotu Feromonu B na políčku.

**-vypfA**  $N$

$N$  je mezi 1 a 100.

Nastavuje kolik jednotek Feromonu A zmizí z políčka během jednoho tahu.

**-vypfB**  $N$

$N$  je mezi 1 a 100.

Nastavuje kolik jednotek Feromonu B zmizí z políčka během jednoho tahu.

**-sbmin**  $N$

$N$  je mezi 1 a 1000.

Nastavuje minální dobu trvání akce, při níž mravenec zdvihá z políčka jídlo a stavební materiál.

**-sbnah**  $N$

$N$  je mezi 0 a 1000.

Tento parametr omezuje shora náhodnou hodnotu, která se přičítá k hodnotě parametru **-sbmin**. Mravenec potřebuje pro zdvihnutí jídla čas, který se rovná součtu u parametru **-sbmin** a náhodné hodnotě z rozsahu  $\langle 0, N \rangle$ .

**-pocstavm**  $N$

$N$  je mezi 0 a 100.

Nastavuje počet stavebního materiálu na mapě.

**-rozstavm**  $N$

$N$  je mezi 1 a 200.

Nastavuje počet stavebního materiálu, potřebného pro rozšíření základny.

**-maxpobyt**  $N$

$N$  je mezi 1 a 500.

Omezuje shora náhodou hodnotu reprezentující čas, který mravenec stráví v mraveništi. Náhodná hodnota leží v rozsahu  $\langle 1, N \rangle$ .

**-konec**  $N$

$N$  je mezi 1 a  $2^{63} - 1$ .

Ukončí v daný čas simulaci. Platí pouze pro konzolovou verzi.

**-vypis**  $N1:[N2:[...]]$

$Nx$  je mezi 1 a  $2^{63} - 1$  a navíc  $Nx < Nx+1$

Statistiky se budou vypisovat pouze v zadané časy. Platí pouze pro konzolovou verzi.

**-rozdil**

Statistiky se budou počítat jako rozdíl mezi jednotlivými výpisy.

Platí pouze pro konzolovou verzi.

**-pocnov**  $N$

$N$  je mezi 1 a 10.

Nastavuje kolik mravenců se vytvoří během jednoho křížení.

**-mnozspot**  $N M$

$N$  je mezi 1 a 50.  $M$  je mezi 1 a  $N$ .

$N$  určuje kolik musí být na základně nadbytečného jídla, aby se mravenci zkřížili.  $M$  určuje kolik jídla se spotřebuje při křížení.

**-nos**  $N$

$N$  je mezi 1 a 10.

$N$  určuje kolik jednotek jídla mravenec unese.

**-dzivot**  $N$

$N$  je mezi 100000 a 1000000.

Nastavuje maximální dobu života mravence. Když mravenec překročí tuto dobu umírá.

**-sims** *SOUBOR*

*SOUBOR* určuje cestu k souboru se simulací který se má načíst.

Načte uloženou simulaci<sup>1</sup> ze souboru.

---

<sup>1</sup>políčka, kalendář jídla, kalendář překážek, poloha základny, počítadla a cesty

**-mras** *SOUBOR*

*SOUBOR* určuje cestu k souboru s mravenci který se má načíst.  
Načte mravence ze souboru.

**-gui**

Spustí GUI. Všechny zadané parametry se budou brát jako výchozí nastavení simulace. Toto nastavení je možné upravit přímo v GUI, nicméně po resetu se obnoví výchozí nastavení.

**-mon** *JMENO:X1xY1:X2xY2:[X1xY1:X2xY2:[X1xY1:X2xY2:[...]]]*

*JMENO* je jméno nového počítadla, musí být unikátní.

*X1*, *X2* jsou mezi 0 a 199.

*Y1*, *Y2* jsou mezi 0 a 99.

Přidá na mapu počítadlo s unikátním jménem. Počítadlo musí mít minimálně jeden záznam. Záznamem se rozumí dvojice souřadnic ((*X1*,*Y1*),(*X2*,*Y2*)). Když mezi nimi projde mravenec, tak počítadlo zvýší svoji hodnotu. Je to vlastně skupina kontrolních bodů monitorujících pohyb mravence. Výstup simulace je doplněn o jména všech počítadel a jejich hodnoty.

Tento parametr má smysl použít víckrát.

**-cst** *JMENO:[POC[-POC[-...]]:[POC[-POC[-...]][:...]]]*

*JMENO* je jméno nové skupiny cest.

*POC* jména počítadel (vytvořených parametrem **-mon**)

Přidá novou skupinu cest. Cesta je reprezentována tvarem p1-p2-p3-p1, kde p1, p2, p3 jsou existující počítadla. Cesty jsou od sebe odděleny dvojtečkou. Cesta má u sebe hodnotu, která určuje počet mravenců, kteří ji využili při úspěšné cestě za potravou. Výstup simulace bude doplněn o jména všech cest a jejich hodnoty.

Tento parametr má smysl použít víckrát.

## 3.2 Ovládání přes GUI

### 1. menu Simulace (*alt + s*)

- **Start** - Spouští simulaci se zvolenou rychlostí.  
*ctrl + s*
- **Pauza** - Zmrazí aktuální stav simulace.  
*ctrl + p*
- **Reset** - Zastaví simulaci a přenastaví všechna data do původního stavu.  
*ctrl + r*
- **Rychlost** - Nastaví rychlost simulace a spustí simulaci.
  - **Nejpomalejší** - Na další tah se čeká 300 ms.  
*ctrl + 1*
  - **Pomalejší** - Na další tah se čeká 200 ms.  
*ctrl + 2*
  - **Normální** - Na další tah se čeká 100 ms.  
*ctrl + 3*
  - **Rychlejší** - Na další tah se čeká 50 ms.  
*ctrl + 4*
  - **Nejrychlejší** - Na další tah se čeká 10 ms (počítač nemusí stihnout překreslení celého okna).  
*ctrl + 5*
  - **Bez Vykreslování** - Okno simulace se nepřekresluje, vývoj simulace lze sledovat pomocí okna Statistika (*ctrl + t*).  
*ctrl + 6*



- **Konec** - Ukončí aplikaci a uvolní program z paměti.  
*ctrl + k*

## 2. menu **Data** (*alt + d*)

- **Základna** - Zobrazí dialog, ve kterém je možné nastavit polohu základny (místa kam mravenci nosí jídlo).  
*ctrl + b*
- **Jídlo na mapě** - Zobrazí dialog, který upravuje oblasti, kde se objevuje jídlo.  
*ctrl + j*
- **Parametry** - Zobrazí dialog s možností nastavení nejrůznějších parametrů.  
*ctrl + v*

### **Jídlo a stavební materiál**

- **Minimální doba zdvihání** - Určuje minimální dobu, kterou mravenec stráví zdviháním jídla a stavebního materiálu.
- **Náhodná doba zdvihání** - Výsledný čas, který mravenec potřebuje pro zdvihnutí jídla nebo stavebního materiálu, se počítá jako součet mezi **Minimální dobou zdvihání** a náhodným číslem z intervalu  $(0, N)$ , kde  $N$  je **Náhodná doba zdvihání**.
- **Počet stavebního materiálu** - Určuje počet jednotek stavebního materiálu na mapě. Pokud je hodnota tohoto parametru nastavena na 0, tak mravenců na mapě nepřibývá, protože nelze zvětšit kapacitu mraveniště.
- **Nutný stavební materiál** - Nastavuje počet stavebního materiálu, který je nutný pro rozšíření kapacity mraveniště o jedna.

- **Energie z jídla** - Nastavuje kolik mravenec získá energie (počet tahů), když se nají.
- **Nové jídlo** - Nastavuje minimální počet jednotek jídla nově vytvořeného zdroje.
- **Koeficient nového jídla** - Při vytváření zdroje nového jídla se k celkovému počtu jednotek jídla na políčku (**Nové jídlo**) přičítá navíc náhodné číslo z intervalu  $(0, N)$ , kde  $N$  je **Koeficient nového jídla**.
- **Pravděpodobnost vytvoření jídla** - Nastavuje pravděpodobnost, že se v příštím tahu na mapě objeví nový zdroj jídla.

### Mravenci

- **Pravděpodobnost mutace** - Nastavuje pravděpodobnost, že při křížení dostane nějaká genová proměnná novou náhodnou hodnotu.
- **Jídlo potřebné pro křížení** - Nastavuje kolik nadbytečného jídla musí být na základně, aby se mravenci zkřížili.
- **Jídlo spotřebované pro křížení** - Nastavuje kolik jídla se spotřebuje při křížení.
- **Počet nových mravenců** - Nastavuje kolik mravenců se vytvoří během jednoho křížení.
- **Počáteční počet mravenců** - Nastavuje kolik mravenců se vytvoří na počátku simulace. Navíc se bude na základně automaticky generovat jídlo, postačující k výživě tohoto počtu jedinců.
- **Nosnost Mravence** - Nastavuje počet jednotek jídla, které mravenec unese.

- **Energetická hranice** - Pokud je energie mravence vyšší, než Energetická hranice, tak se mravenec nenají. Mravenec se navíc na základě tohoto parametru a genetické výbavy rozhodne, kdy se začne vracet do mraveniště.
- **Doba života** - Jakmile přesáhne stáří mravence tuto hodnotu, tak mravenec umírá.
- **Pobyt v mraveništi** - Ovlivňuje dobu, kterou mravenec stráví v mraveništi při doplňování energie a vykládání jídla nebo stavebního materiálu. Doba se vybírá jako náhodné číslo v rozmezí  $\langle 1, N \rangle$ .

### **Feromony**

- **Intenzita Feromonu A** - Nastavuje kolik jednotek Feromonu A se objeví na políčku po průchodu mravence vylučujícího Feromon A.
- **Intenzita Feromonu B** - Nastavuje kolik jednotek Feromonu B se objeví na políčku po průchodu mravence vylučujícího Feromon B.
- **Maximum Feromonu A** - Určuje maximální počet jednotek Feromonu A na políčku.
- **Maximum Feromonu B** - Určuje maximální počet jednotek Feromonu B na políčku.
- **Rychlost vypařování Feromonu A** - Počet jednotek Feromonu A, které se vypaří za jeden tah.
- **Rychlost vypařování Feromonu B** - Počet jednotek Feromonu B, které se vypaří za jeden tah.

- **Kalendář Jídla** - Zobrazí dialog, kde se dá nastavit, jak se budou měnit zdroje jídla v závislosti na čase.  
*ctrl + g*
- **Kalendář Překážek** - Zobrazí dialog, kde se dá nastavit, jak se budou měnit překážky v závislosti na čase. Lze zde měnit průchodnost (0-3) jednotlivých oblastí během simulace.  
*ctrl + z*
- **Počítadla a cesty** - Zobrazí dialog, kde se nastavují body sloužící pro monitorování simulace. Počítadla jsou skupiny spojnic mezi dvěma body, které počítají kolik mravenců jimi prošlo a cesty jsou skupiny posloupností počítadel, která zjišťují, kolik mravenců využilo danou cestu skupinu cest ke zdroji jídla.  
*ctrl + y*
- **Překážky** - Zde se volí, jaké překážky se umístí na plochu po stisku levého tlačítka.
  - **Neprůchodné** - Překážka je neprůchodná.  
*ctrl + q*
  - **Těžko Průchodné** - Průchod zabere mravenci tři tahy.  
*ctrl + w*
  - **Hůře Průchodné** - Průchod zabere mravenci dva tahy.  
*ctrl + e*
- **Nekonečný svět** - Určuje, co se stane, když se mravenec dostane k okraji plochy. Pokud je zatržena, tak se mravenec po přezení okraje ocitne na opačné straně plochy. Když není, tak se mravenec chová jakoby by byly okraje neprůchodné.  
*ctrl + i*
- **Paměť mravence** - Zapíná / Vypíná paměť mravence. Mravenec si za normálních okolností přesně pamatuje, cestu kterou přišel z mraveniště (bez smyček). Pokud vypneme paměť mravence, tak

se mravenec pokusí dostat zpátky do mraveniště pomocí feromonové stopy, kterou vylučoval cestou od mraveniště.

*ctrl + m*

- **Načti Mravence** - Načte mravence z binárního souboru.  
*ctrl + n*
- **Ulož Mravence** - Uloží mravence do binárního souboru.  
*ctrl + u*
- **Načti Simulaci** - Načte uloženou simulaci ze souboru. Simulací se rozumí políčka, kalendář jídla, kalendář překážek, poloha základny, počítadla a cesty.  
*ctrl + c*
- **Ulož Simulaci** - Uloží simulaci do souboru.  
*ctrl + d*
- **Edituj Mapu** - Resetuje simulaci a zapne/vypne mód pro editaci mapy.  
*ctrl + f*

### 3. menu **Zobrazení** (*alt + z*)

- **Statistika** - Zapíná / Vypíná zobrazování okna Statistika.  
*ctrl + t*
- **Data k zobrazení** - Zde se dá zvolit, která data se budou zobrazovat v okně Statistika. Hodnoty se budou zobrazovat ve stejném pořadí, jaké je v menu.
  - **Čas** - Zobrazí čas.  
*ctrl + 7*
  - **Počet mravenců** - Zobrazí počet mravenců.  
*ctrl + 8*

- **Množství jídla** - Zobrazí množství jídla v mraveništi.  
*ctrl + 9*
- **Kapacitu mraveniště** - Zobrazí maximální počet mravenců, jenž je schopné mraveniště pojmout.  
*ctrl + 0*
- **Počítadla** - Zobrazí jména počítadel a kolik mravenců už jimi prošlo.  
*ctrl + -*
- **Cesty** - Zobrazí jména cest a kolik mravenců je již použilo.  
*ctrl + .*
- **Smyčky** - Zobrazí počet cest se smyčkami.  
*ctrl + =*
- **Celkem cest** - Zobrazí celkový počet cest.  
*ctrl + ,*
- **Zobrazit mřížku** - Zapíná / Vypíná zobrazování mřížky.  
*ctrl + x*
- **Zobrazit feromony** - Zapíná / Vypíná zobrazování feromonů.  
*ctrl + l*
- **Zobrazit Počítadla** - Zapíná / Vypíná zobrazování počítadel.  
*ctrl + /*

#### 4. menu Nápověda (*alt + n*)

- **O programu** - Zobrazí informace o programu.  
*ctrl + h*

## 5. Tlačítka myši

- **Levé** - Přidává / Odebírá překážky.
- **Prostřední** - Přidává oblasti, ve kterých se objevuje jídlo. V editačním módu pak přidává/odebírání oblasti překážek.
- **Pravé** - Přidává jídlo.

# Kapitola 4

## Porovnání se skutečnými mravenci

V této části se nachází porovnání navrženého modelu s chováním skutečných mravenců popsaném v [1].

### 4.1 Experiment se skutečnými mravenci

#### 4.1.1 Popis prostředí

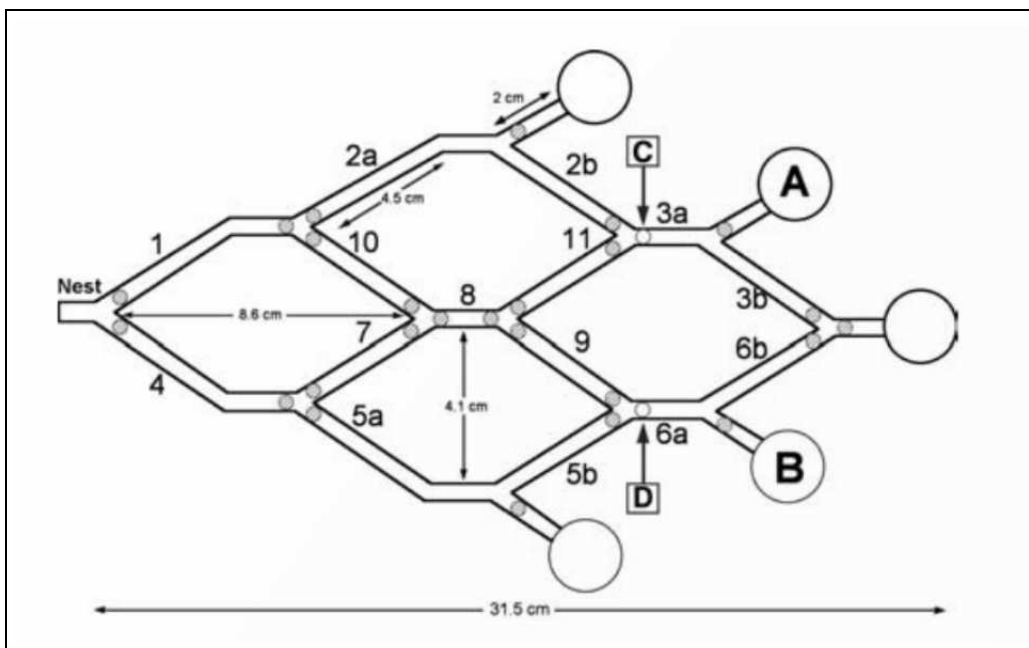
Experiment se odehrál v prostředí uměle vytvořeného bludiště. Bludiště se skládalo ze čtyř identických kosočtverců složených dohromady a z pěti kruhových dutin, viz Obrázek 4.1.

Pro pokus byli použiti Argentinští mravenci *Linepithema humile*. Bylo vytvořeno 10 pokusných kolonií, každá obsahující cca 2000 jedinců. Všichni mravenci byli posbíráni mimo mraveniště a můžeme je tudíž považovat za sběrače. Každá kolonie byla umístěna v plastovém kontejneru a ponechána 4 dny bez jídla.

#### 4.1.2 Popis experimentů

Na počátku experimentu se kolonie spojila s bludištěm plastovým mostem. Mravenci museli projít přes tento most, pak skrz bludiště až ke zdroji potravy a zpátky. Jídlo bylo umístěno v polovině experimentů do dutiny A,





Obrázek 4.1: Prostředí experimentu s reálnými mravenci - Obrázek z [1]

Cesty ke zdroji A	Cesty ke zdroji B	Délky cest
1-2a-2b-3a 1-10-8-11-3a 4-7-8-11-3a	4-5a-5b-6a 1-10-8-9-6a 4-7-8-9-6a	s - 21,5 cm krátké
4-5a-5b-9-11-3a 4-7-10-2a-2b-3a 4-5a-5b-6a-6b-3b 4-7-8-9-6a-6b-3b 1-10-8-9-6a-6b-3b	1-2a-2b-11-9-6a 1-10-7-5a-5b-6a 1-2a-2b-3a-3b-6b 1-10-8-11-3a-3b-6b 4-7-8-11-3a-3b-6b	m - 30,5 cm střední
1-2a-2b-11-9-6a-6b-3b 1-10-7-5a-5b-6a-6b-3a 1-10-7-5a-5b-9-11-3a	4-5a-5b-9-11-3a-3b-6b 4-7-10-2a-2b-3a-3b-6b 4-7-10-2a-2b-11-9-6a	l - 39,5 cm dlouhé
4-7-10-2a-2b-11-9-6a-6b-3b	1-10-7-5a-5b-9-11-3a-3b-6b	vl - 48,5 cm velmi dlouhé

Tabulka 4.1: Možné cesty k potravě

v druhé polovině do dutiny B. Po každém experimentu bylo bludiště řádně očištěno. Každý experiment trval 60 minut. V síti existuje dvanáct možných cest (bez smyček) jak se dostat skrz bludiště ke zdroji potravy, viz Tabulka 4.1.

Všech deset kolonií bylo testováno třemi různými způsoby:

1. Prostředí simulace se neměnilo.
2. Prvních třicet minut, měli mravenci přístup do všech částí bludiště. Poté se zablokovala chodba těsně u zdroje potravy. V případě, když je zdroj jídla v dutině A, zatarasí se 3a. Pokud je v dutině B, tak 6a.
3. Prvních třicet minut mohli mravenci využívat pouze jednu cestu ke zdroji potravy. Pokud bylo jídlo uloženo v dutině A, to byla cesta 1-10-7-5-9-11-3a. V případě dutiny B 4-7-10-2-11-9-6a. Poté měli mravenci přístup do celého bludiště.

### 4.1.3 Výsledky pokusů

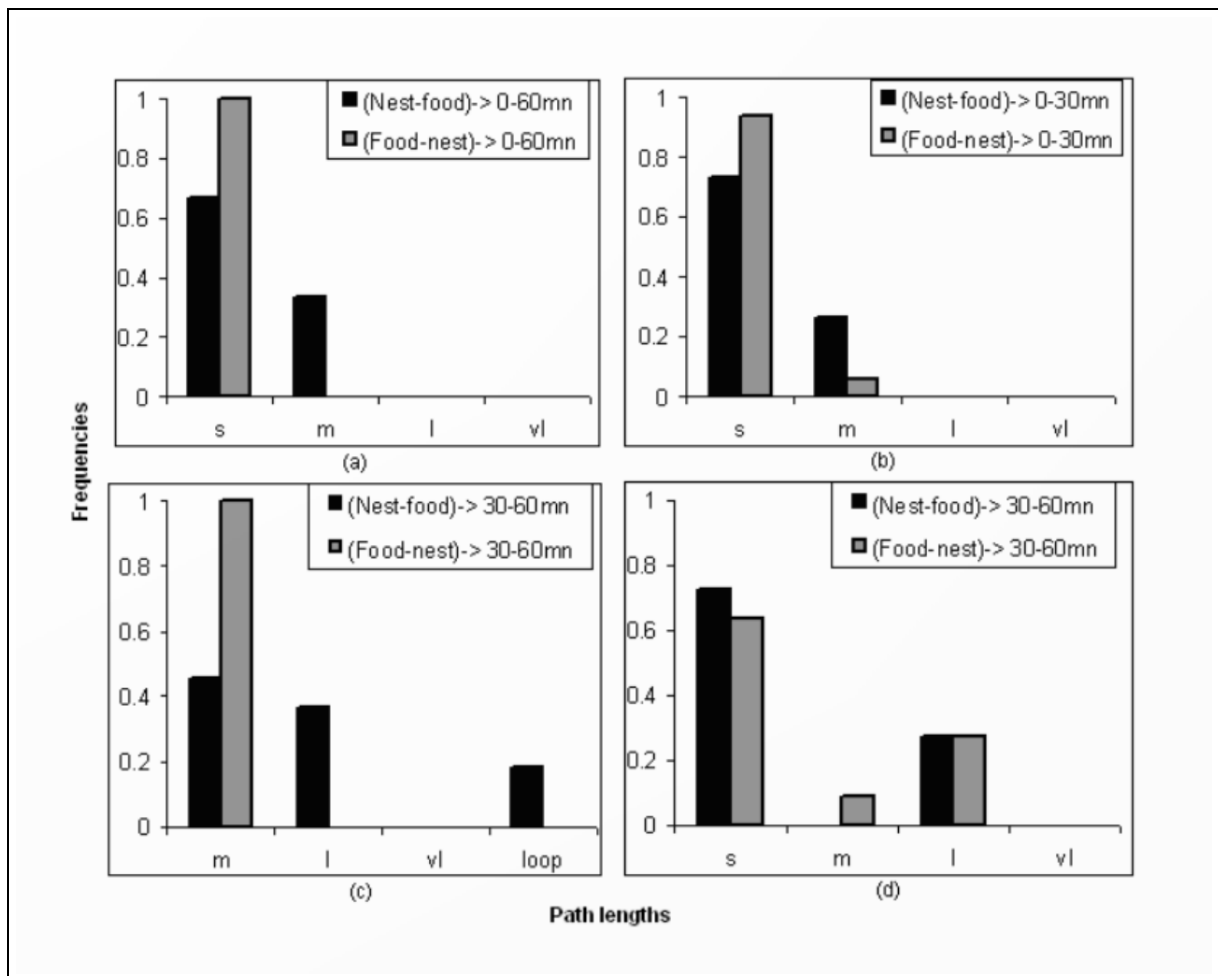
#### Reální mravenci

Ve většině experimentů šli mravenci ke zdroji jídla nejkratší cestou. Ve všech experimentech zvolili nejkratší cestu od zdroje jídla do mraveniště. Nicméně v druhé části experimentů po zablokování cest nebyly některé kolonie schopné najít cestu ke zdroji jídla. Mravenci se v takovém případě dostali do smyčky, ze které se už nedostali. Některým koloniím se také po odblokování cest nedařilo nalézt nejkratší cestu k jídlu. Během simulace využívaly jen cestu, která byla volná v první části experimentu. Jedna kolonie dokonce začala využívat více cest naráz včetně původní, viz Obrázek 4.2.

U mravenců byly také naměřeny další zajímavé charakteristiky:

- Průměrná rychlost mravence -  $1.10 \text{ cm/s}$
- Frekvence otáčení o  $180^\circ$  stupňů při cestě ke zdroji jídla - 0.15
- Frekvence otáčení o  $180^\circ$  stupňů při cestě do mraveniště - 0.10
- Průměrná doba pobytu u zdroje jídla. -  $185.30 \text{ s}$

- Průměrný Maximální počet mravenců v bludišti - 100 jedinců



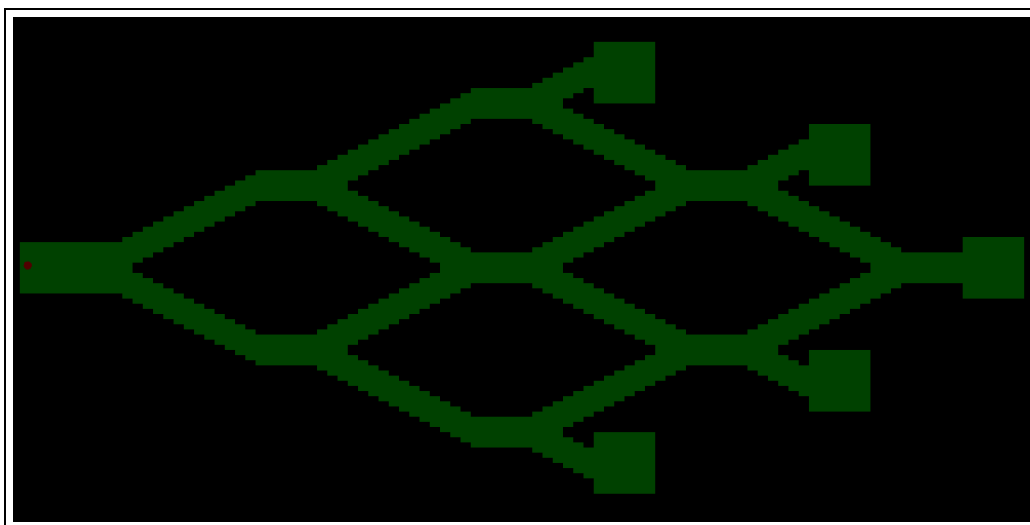
Obrázek 4.2: Výsledky experimentu s reálnými mravenci - Obrázek z [1]  
 (a) Neomezený přístup do všech míst celých 60 minut. (b)-(c) Zablokování cesty: frekvence před (b) a po (c) zablokování cesty ve 30. minutě. (d) Po uvolnění všech cest, pouze druhá polovina experimentu.

## 4.2 Simulace

### 4.2.1 Popis prostředí

Mapu jsem vytvářel v editačním módu programu (viz 3.2). Jeden centimetr by měl odpovídat zhruba šesti políčkům. Kvůli častému ucpávání cest poblíž vstupu do mraveniště jsem se rozhodl šířku cest zdvojnásobit, viz Obrázek 4.3.

Mravenci měli k dispozici 3764 přístupných políček.



Obrázek 4.3: Prostředí vytvořené v editačního módu

### 4.2.2 Popis simulace

Program nejprve vytvoří určitý počet mravenců s náhodnou genetickou výbavou. Tito mravenci jsou ale nevhodní pro porovnání, protože se u nich ještě nevyvinula dobrá strategie pro hledání a transport potravy. Kvůli tomu je nutné si nejprve připravit vyzrálejší populaci. Tato populace byla vytvořena v grafickém módu programu postupným přesouváním zdroje jídla a přidáváním překážek.

Vzhledem k průměrné rychlosti reálných mravenců pohybujících se na mapě (viz 4.1.3) bude simulace trvat 25000 tahů. Počet jedinců bude 100. Na mapě se nebude vyskytovat stavební materiál. Tím docílíme toho, že mravenců nebude na mapě přibývat. Mravenci stráví u zdroje jídla 900 až 1300 tahů. V mraveništi pak nebudou déle než 400 tahů.

Každá simulace probíhala 100-krát a byla spouštěna s parametry:

- **pocm** 100
- **konec** 25000
- **vypis** 12500:25000
- **rozdil**
- **sbmin** 900
- **sbnah** 400
- **maxpobyt** 400
- **pocstavm** 0

Vlastní pokus měl stejnou strukturu, jako pokus s opravdovými mravenci.

Výsledky se dělí:

1. Dle typu simulace:

**Obyčejná** - prostředí se neměnilo.

**Zablokování** - v půlce simulace se zablokovaly nejkratší cesty do mraveniště a mravenci byli nuceni použít delší cestu.

**Zprůchodnění** - na počátku bylo možné použít pouze jednu dlouhou cestu a v půlce simulace se všechny cesty zprůchodnily.

2. Dle umístění zdroje potravy (viz Obrázek 4.1):

**A** - jídlo bylo v dutině A

**B** - jídlo bylo v dutině B

3. Dle času:

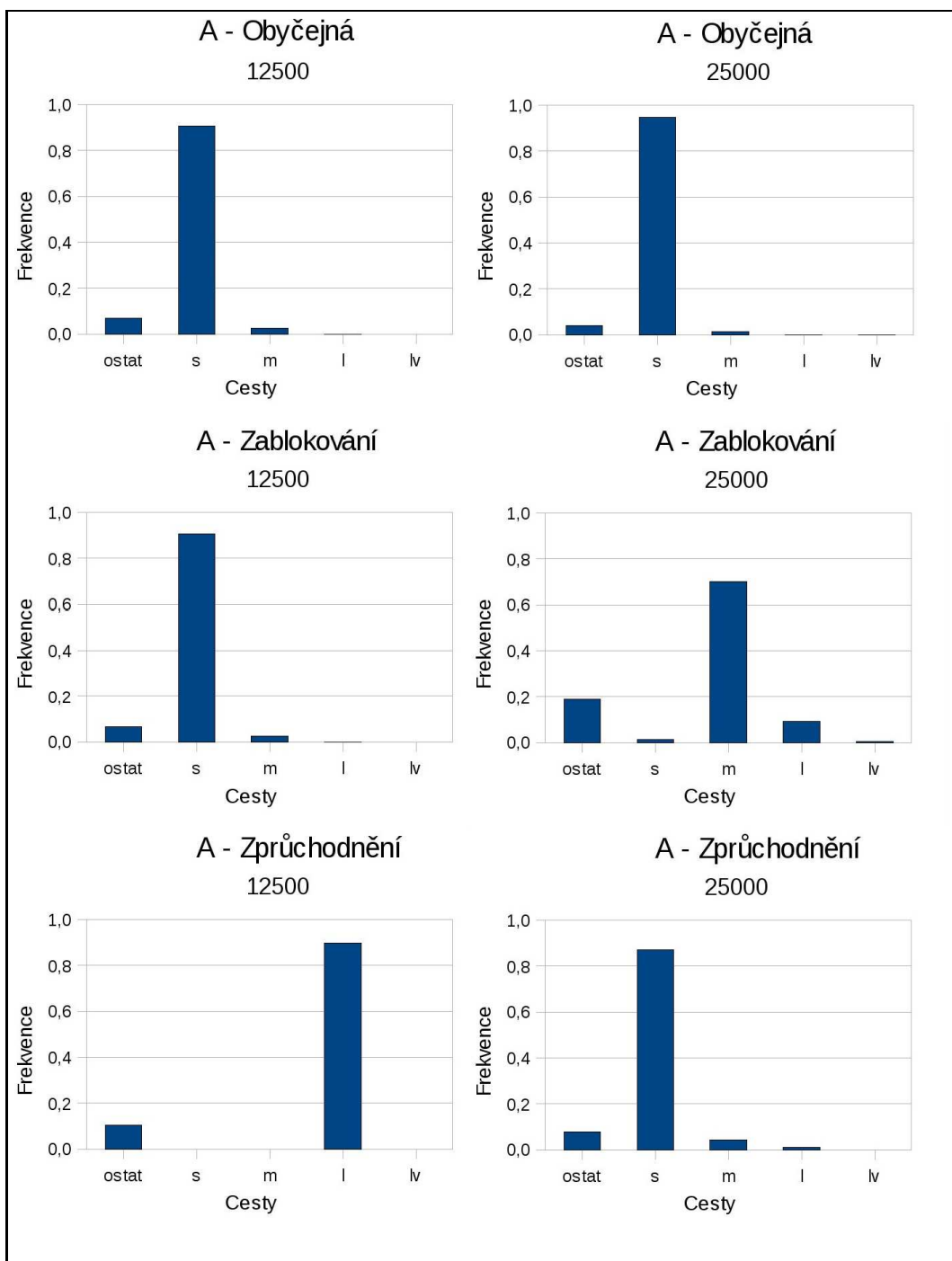
**12500** - první část simulace.

**25000** - druhá část simulace.

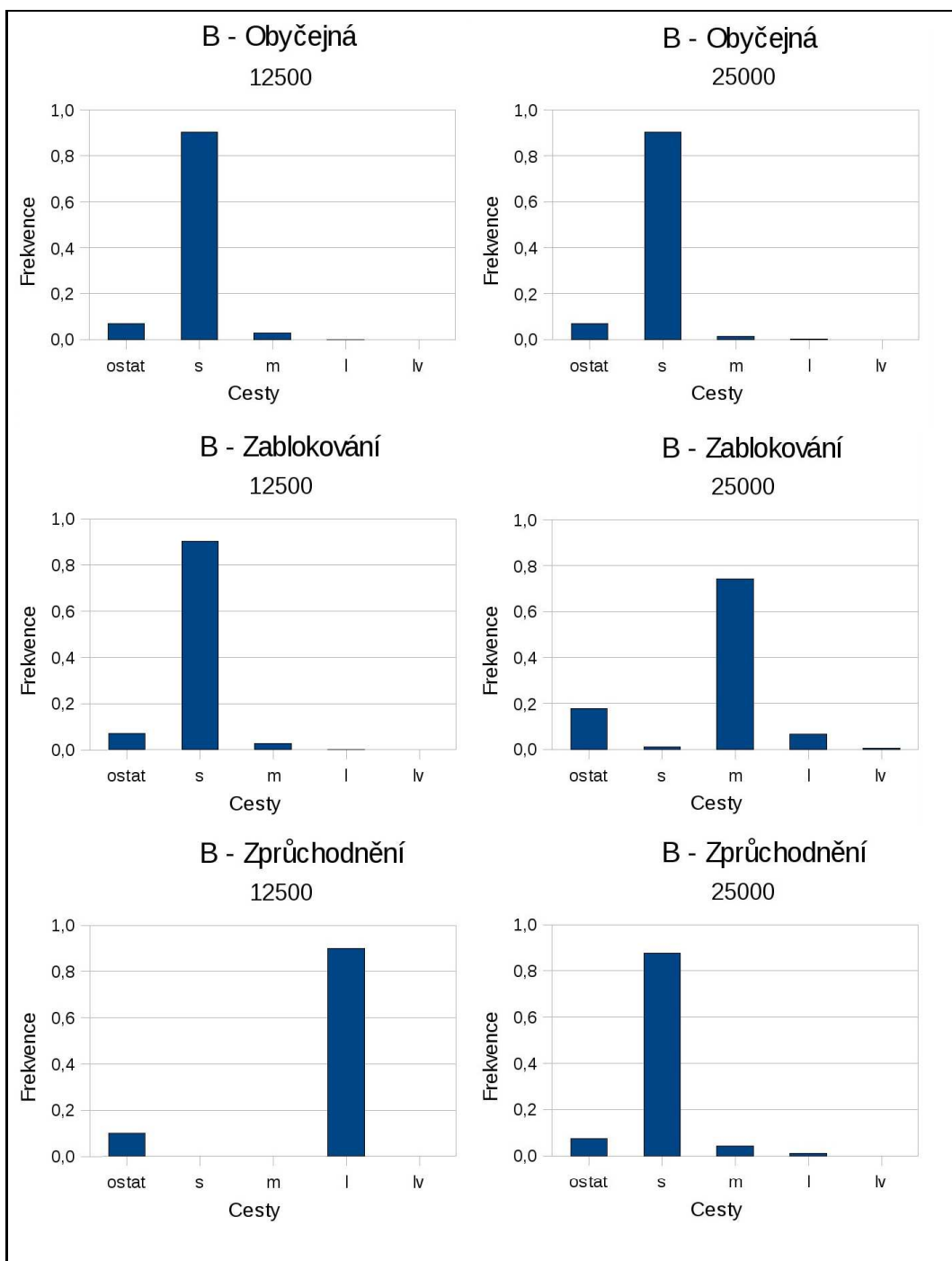
Dále byl také sledován celkový počet nalezených cest ke zdroji jídla a celkový počet cest ke zdroji jídla se smyčkami.

	Průměrně našli cest	Z toho se smyčkami
A Obyčejná 1. část	238,85	96,15
A Obyčejná 2. část	300,25	83,23
A Zablokování 1. část	238,21	96,14
A Zablokování 2. část	47,97	34,14
A Zprůchodnění 1. část	125,38	100,49
A Zprůchodnění 2. část	221,03	83,49
B Obyčejná 1. část	238,99	96,47
B Obyčejná 2. část	298,41	85,91
B Zablokování 1. část	238,26	97,72
B Zablokování 2. část	54,07	37,87
B Zprůchodnění 1. část	125,96	99,19
B Zprůchodnění 2. část	233,75	86,96

Tabulka 4.2: Průměrné počty cest a smyček



Obrázek 4.4: Výsledky pokusu s jídlem v dutině A



Obrázek 4.5: Výsledky pokusu s jídlem v dutině B



### 4.2.3 Výsledky

Je snadno pozorovatelné, že pokud jsou simulace stejného typu liší se pouze v umístění zdroje jídla, tak jsou jejich grafy takřka totožné, viz Obrázky 4.4 a 4.5.

Při pokusech se zablokováním cesty se i ve druhé části experimentů objevuje menší množství cest délky  $s$ , které by měly být zablokované. Je to dáno tím, že se někteří mravenci vyskytovali již za nově přidanou překážkou, ale ještě nestihli nalézt jídlo. Ve statistikách se čítač u cest zvyšuje až v momentě, kdy mravenec najde jídlo. Tím pádem se do statistik započítala krátká cesta, ačkoliv v tu dobu už nebyla průchodná.

Ve výsledcích je poměrně často zastoupena skupina cest **ostat**, reprezentující ostatní cesty<sup>1</sup>. Z cest jsou sice při přidávání do statistik odstraňovány cykly. Nicméně pokud mravenec vstoupí do počítadla a vystoupí z něj na stejné straně, kudy vešel, pak se přidané počítadlo nepovažuje za smyčku a tudíž není při přidávání do statistik odstraněno. Cesta s takovýmto "hrotem" je řazena mezi ostatní. Největší výskyt těchto cest je v situaci po zablokování krátkých cest ke zdroji jídla.

## 4.3 Srovnání

Skuteční mravenci používají při cestě ke zdroji potravy a zpět různé cesty. Naproti tomu simulovaní mravenci chodí zpátky do mraveniště po témeř totožné cestě, ze které se jen odstraní smyčky. Veškerá srovnávání se proto budou týkat pouze cesty k jídlu.

### 4.3.1 Obyčejná

Výsledky se zde nejsou zas až tak rozdílné. Vypadá to, že mravenci ze simulace jsou trochu lepší, vezmeme-li ale v potaz relativně vysoké množství smyček, které se započítávaly jako regulerní cesta, tak se jejich výkonnost nezdá až tak oslnivá. Nicméně v obou případech se jasně ukazuje, že mra-

---

<sup>1</sup>Dále v textu jsou také označovány jako nezařaditelné.

venci preferovali nejkratší možnou variantu.

### **4.3.2 Zablokování**

V této části experimentů měli mravenci v obou případech poměrně dost problému najít optimální cestu. Opravdoví mravenci chodili celkem často po dlouhých cestách a navíc některé kolonie vůbec nebyly schopné najít cestu ke zdroji a cesty končily ve smyčce. Mravenci ze simulace také nijak moc neoslhli. Dlouhé cesty vybírali sice je zřídka, ale během pokusu využili nezanedbatelné množství nezařaditelných cest (viz 4.2.3). Navíc se zde cesty bez smyček vyskytovaly opravdu po skromnu.

### **4.3.3 Zprůchodnění**

Skuteční mravenci si v této části pokusu vedli relativně slušně. Nicméně některé kolonie nebyly schopné najít nejkratší cestu do mraveniště a držely se známé vyšlapané cestičky.

Mravenci ze simulace také podávali velice dobré výkony. Sice měli v první části simulace vysoký podíl smyček, ale ve druhé části pro ně nebyl problém najít a využívat nejkratší cestu. Jejich graf z druhé části je hodně podobný grafům simulace s neměnným prostředím.

# Kapitola 5

## Závěr

### 5.1 Diskuze

Po porovnání výsledků by se dalo říct, že se mravenci ze simulace umí lépe přizpůsobit změnám prostředí. Je to dáno tím, že celkem často hledají nové cesty a snaží se více objevovat. Toto chování by se jistě dalo ovlivnit tak, že by se změnil poměr objevitelů a nosičů. Otázkou ovšem zůstává, jak tento poměr vypadá a jaký má vliv na efektivitu sběru jídla.

Reální mravenci se naopak snaží držet jedné cesty, což může mít neblahý vliv na efektivitu. Na to doplatily především kolonie, které začaly chodit v kruhu v druhé části experimentu se zablokováním cest. Řekl bych, že by větší množství "objevitelů" mohlo těmto koloniím pomoci k lepším výsledkům. Nicméně pokud vezmeme v úvahu nástrahy reálného světa, musíme připustit že mravenec, který se ocitne na cizím jídelníčku, už moc toho jídla neodnese a osamělý objevitel je pro predátora velmi snadná kořist.

Při pokusu s umělými mravenci se často stávalo, že mravenec při cestě do mraveniště využil cestu, která obsahovala smyčku. Navíc se zde vyskytovalo také poměrně velké množství nezařaditelných cest. Tyto jevy jsou dle mého názoru způsobeny občasným "ucpáním" cest, kdy mravenci nezbyde nic jiného, než se vrátit a tím vytvoří smyčku. Nezařaditelné cesty by šly ze statistik odstranit lepší implementací počítadel, která by lépe rozlišovala průchody mravenců přes políčka počítadla. Tento oprava je plánována do budoucích verzí programu.

## 5.2 Závěr

Cílem práce bylo navržení a vytvoření modelu napodobujícího chování mravenců při sběru potravy. Oproti původnímu plánu implementovaný model obsahuje navíc rozdělení hodnocení mravenců na nosiče a objevitele (v původním plánu byli pouze nosiči) a nekonečný svět (viz 3.1). Navíc se kvůli možnosti srovnání s reálnými mravenci musel do aplikace zakomponovat celý systém monitorování cest a počítačů. Přibyly také řada drobnějších doplňujících nastavení například nastavitelná doba, kterou mravenec stráví v mraveništi, či nastavitelná doba, kterou stráví nakládáním jídla. Tyto vlastnosti hodně přispěly k vytvoření složitějšího prostředí. Porovnání s výsledky experimentů z [1] ukázalo, že se náš model svým chováním přiblížil k chování skutečných mravenců, nicméně k dokonalosti mu ještě něco chybí.

Model postrádá například:

- Lepší implementaci počítačů (viz 2.10.4)
- Odstranění nezařaditelných cest (viz 4.2.3)
- Přidání nastavení poměru mezi nosiči a objeviteli
- Možnost přidání více vchodů do mraveniště
- Nová strategie mravenců při návratu do mraveniště, tak aby více odpovídala skutečnosti.
- atd.

Práce na projektu byla časově náročnější, než jsem původně očekával. Nicméně jsem se během práce dověděl spoustu zajímavých věcí o etologii mravenců a doufám, že jsem aspoň trochu přispěl k objasnění jejich komplexní etologie.

# Kapitola 6

## Přílohy

### 6.1 CD

CD obsahuje:

- Zdrojové kódy programu psané v jazyce Java
- Přeložený program
- Ebuild pro Gentoo
- Vygenerovaný javadoc

# Literatura

- [1] Karla Vittori, Jacques Gautrais, Aluizio F.R. Arajúo, Vincent Fourcassié and Guy Theraulaz: *Modeling Ant Behaviour Under a Variable Environment*, ANTS Workshop, 2004.
- [2] Karel Zvára, Josef Štěpán: *Pravděpodobnost a matematická statistika*, MATFYZPRESS, 2006.