



MATEMATICKO-FYZIKÁLNA FAKULTA
UNIVERZITY KOMENSKÉHO
KATEDRA UMELEJ INTELIGENCIE

Andrej Lúčný

Emergentné správanie v kolóniách agentov

Diplomová práca

BRATISLAVA

1994

Prehlasujem na svoju česť, že predkladanú diplomovú prácu som vypracoval samostatne a všetku použitú literatúru uvádzam v zozname.

Bratislava, marec 1994

Touto cestou si dovoľujem poďakovať Doc. RNDr. Jozefovi Kelemenovi, CSc. za starostlivé vedenie, za poznámky a rady k spracovávanej oblasti a za sprístupnenie potrebnej literatúry.

Ďakujem aj mojim rodičom za ich všestrannú pomoc počas celého štúdia.

Ej, Zajko, veď som ja už dosť múdra, len to mi ešte povedz, ako vykopali ľudia jamu do vody pre tento dom?

Jozef Cíger Hronský:
Smelý Zajko v Afrike

Kapitola 1

Úvod

Vo svojom okolí pozorujeme mnoho zložitých systémov, ktorým pripisujeme schopnosti, ktoré iné systémy a to hlavne systémy ľuďmi stvorené (t.j. stroje) nemajú. Tieto schopnosti im pripisujeme na základe pozorovania ich správania, ktoré sa výrazne líši od správania, ktoré dokážeme vštepiť súčasným strojom. Zdroj týchto schopností a teda i možnosť umelo ich vytvoriť nám ostávajú skryté a to dráždi našu ctižiadosť.

Svedčí i tom i fakt, že ľudia si na vysvetlenie prítomnosti týchto schopností vytvorili predstavu o nehmotnej substancii, ktorá tieto schopnosti systému dodáva - predstavu duše.

Keď pozorujeme nejaký systém v prírode, pozorujeme ho spravidla na určitej úrovni (na úrovni atómov, buniek, orgánov, organizmov, ...), na ktorej sa skladá z podobných jednotiek, medzi ktorými prebieha nejaká interakcia. Na základe javov pozorovaných na tejto úrovni však nedokážeme vysvetliť všetky javy na vyšších úrovniach.

Živočíchy sa vyznačujú schopnosťou riešiť problémy, ktoré pred ne stavia potreba prispôbiť sa prostrediu v ktorom žijú, prípadne si ho zmeniť, aby si zabezpečili prežitie. Vieme, že tento proces prebieha ako presun elektrického potenciálu v nervových bunkách týchto živočíchov, ale na základe toho nie sme schopní povedať o danom jave nič podstatné.

Dobrym predmetom na pozorovanie sú obligátne spoločenstvá (mravce, termity, včely, ...). Jedno také spoločenstvo tvorí celok, skladajúci sa z jedincov, medzi ktorými prebieha určitý druh interakcie. Máme tu teda dve úrovne: úroveň jedincov a úroveň spoločenstva. Na nižšej úrovni vieme dosť dobre opísať, čo taký jedinec dokáže, vrátane spôsobu interakcie, no na základe toho nedokážeme vysvetliť schopnosti spoločenstva na úrovni vyššej.

Interakcia medzi jedincami obligátneho spoločenstva má rôzne formy, od priameho kontaktu cez komunikáciu feromónmi až po rituálne pohyby.

Existuje istá analógia medzi strojmi a obligátnymi spoločenstvami. Týka sa distribuovaných robotických systémov. Vynára sa otázka, či by viac rovnakých robotov mohlo spolu utvoriť systém so schopnosťami, ktoré nemá samostatný robot. Abstrahujúc od stelesnenia, dostávame sa k pojmom kolónia a agent, ktoré popisujú spoločenstvo a jedinca v tomto prípade.

1.1 Agent

Pod **agentom** rozumieme niečo¹, čo neustále vníma svoje prostredie, na základe toho volí akcie, ktoré má v tomto prostredí vykonať, aby dosiahol vopred stanovený cieľ a následne tieto akcie vykonáva.

Existujú dva základné prístupy k spôsobu voľby akcií. Prvý, historicky starší, sa nazýva **deliberatívny** a spočíva v tom, že voľba akcií prebieha na základe vnútornej reprezentácie prostredia a uvažovania nad výsledkami volených akcií (agent sa „rozhoduje“). Druhý spôsob sa nazýva **nedeliberatívny** a spočíva v tom, že voľba akcií prebieha na základe reakcie na stav prostredia a agenta (agent „reaguje“). Podrobnejšiu charakterizáciu oboch prístupov uvádza [Ferguson 92].

Podľa spôsobu voľby akcií potom delíme agentov na deliberatívnych a nedeliberatívnych. Nedeliberatívnych agentov, ktorých stav sa nezakladá na vnútornej pamäti (t.j. nemajú vnútorný stav), nazývame (**čisto**) **reaktívnymi agentami**.

V tejto práci sa budeme zaoberať čo najjednoduchšími nedeliberatívnymi agentami. Preto v ďalšom budeme pod agentom vždy rozumieť nedeliberatívneho agenta.

1.2 Kolónia

Pod **kolóniou** rozumieme viacej agentov, medzi ktorými prebieha nejaká interakcia (prípadne až komunikácia).

1.3 Emergentné správanie

Kolónia umiestnená do nejakého prostredia predstavuje systém, v ktorom vzniká zaujímavé globálne správanie. To je výsledkom správania sa agentov a interakcie, ktorá medzi nimi prebieha. Správanie určitej kolónie sa spravidla veľmi ťažko predpovedá. Rovnako ťažko sa hľadá pre určité správanie kolónia, ktorá ho produkuje. Vzhľadom na tieto skutočnosti nazývame toto správanie emergentným ([Forest 91]).

¹ Väčšinou sa na tomto mieste spomína slovo proces ([Doran 92], [Goodwin 93]).

Z hľadiska umelej inteligencie je **emergentné správanie** zaujímavé tým, že v istých prípadoch pripomína správanie sa živých systémov.

1.4 Cieľ práce

Cieľom tejto práce je preskúmať možnosti produkovania emergentného správania kolóniou nedeliberatívnych agentov čo najjednoduchšej architektúry. Tento výskum spočíva v realizácii niekoľkých experimentov, pri ktorých je možné pozorovať emergentné správanie.

Experiment spočíva v simulácii aktivít zvolenej kolónie agentov vo zvolenom prostredí. Hlavným kritériom na posúdenie, či pri experimente dochádza k emergentnému správaniu bude intuícia, no budú ju sprevádzať i merania a vyhodnocovania určitých kvantitatívnych ukazovateľov. K prevádzaniu týchto experimentov je nevyhnutné

- stanoviť architektúru agentov a spôsob jej zapisovania
- stanoviť, ako prebiehajú aktivity kolónie agentov v nejakom prostredí v čase a spôsob ich modelovania
- stanoviť spôsob kvantitatívneho vyjadrenia prítomnosti alebo absencie emergentného správania
- implementovať programové prostredie umožňujúce realizovať spomínané experimenty a merať pritom vybrané kvantitatívne ukazovatele.

1.5 Predchádzajúce práce

Táto práca nadväzuje na práce v oblasti tzv. novej umelej inteligencie. Novým prístupom k umelému vytvoreniu inteligentných systémov na základe metódy zdola hore sa ako prví zaoberali R. A. Brooks a J. H. Connell. ([Brooks 86], [Brooks 89]). Brooks zaviedol v súvislosti s umelou inteligenciou pojem emergencie a navrhol použiť pri konštrukcii jedného agenta subsumpčnú architektúru založenú na dekompozícii aktivitou². Agent sa tak stal kompozíciou modulov, predstavujúcich jednotlivé správania (aktivity), medzi ktorými prebieha určitá komunikácia. Connell prišiel s predstavou čisto reaktívneho agenta. M. J. Mataric a L. E. Parker sa zaoberali komunikáciou medzi agentami a ich spájaním do kolónií ([Mataric 92], [Parker 92a]). Modelovaním aktivít agentov v prostredí sa zaoberali R. Goodwin a P. Maes ([Goodwin 93], [Maes 89]).

² Tento návrh a systémy realizované v jeho intenciách sú podrobnejšie opísané v [Mlichová 93]

Podčiarknuté a zrátané: dospelo sa k presne vymedzenej predstave kolónie zloženej z interagujúcich agentov, ktoré sú zložené z interagujúcich modulov, čím vznikol systém s tromi úrovňami (obrázok 1).



Obrázok 1

Výsledky dosiahnuté v tejto oblasti sú pomerne málo teoreticky podložené. Hlavný problém teoretických modelov spočíva v absencii prirodzeného prostredia, ktoré je zdrojom vstupov do agentov a ktoré inšpiruje ich správanie ([Forest 91]). Teoretickým prístupom sa zaoberali J. Kelemen a A. Kelemenová, ktorí sa v [Kelemen - Kelemenová 92] snažia definovať štandardný gramatický formalizmus na opis agentov a ich správania vrámci kolónie opierajúci sa o štandardné prostriedky formálnych jazykov a gramatík. [Kelemen 93] predstavuje pokus o spojenie tohto formalizmu s konkrétnymi technickými experimentami v oblasti kolektívnej robotiky. [Baník 94] prezentuje alternatívny formalizmus, v ktorom sa využíva aparát abstraktných strojov. Všetky tieto práce vychádzajú zo spoločnej filozofickej predstavy. Držať sa tejto predstavy sa snažíme i v tejto práci. To nás núti urobiť niekoľko obmedzení, ktoré sa týkajú hlavne jednoduchosti štruktúry agentov.

Kapitola 2

Architektúra agenta

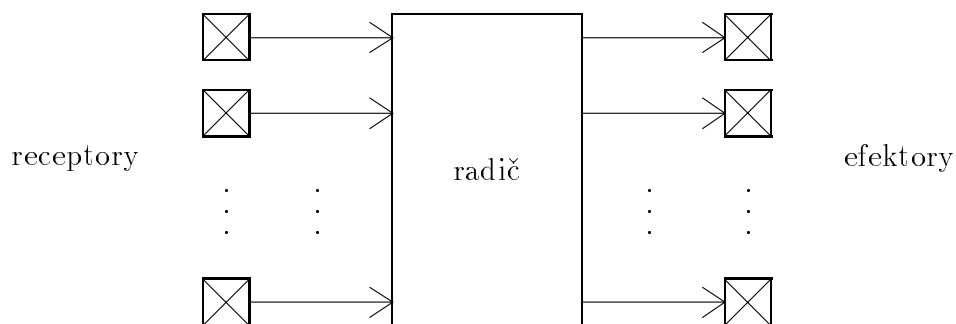
2.1 Základné časti agenta

Agent sa skladá z dvoch základných častí: z mechanizmu a z radiča ([Goodwin 93]).

Mechanizmus predstavujú **receptory** (senzory) a **efektory** (aktuátory). Receptory slúžia na vnímanie prostredia, v ktorom sa agent nachádza. Efektory naopak zabezpečujú vykonanie zvolených akcií v tomto prostredí.

Radič implicitne „pozná“ cieľ (úlohu), prijíma signály z receptorov a vysiela riadiace signály na efektory, na základe ktorých vykonávajú efektory v prostredí akcie, ktoré vedú k naplneniu cieľa.

Radič môže mať rôznorodú architektúru, ktorá je rozhodujúcim faktorom architektúry agenta.

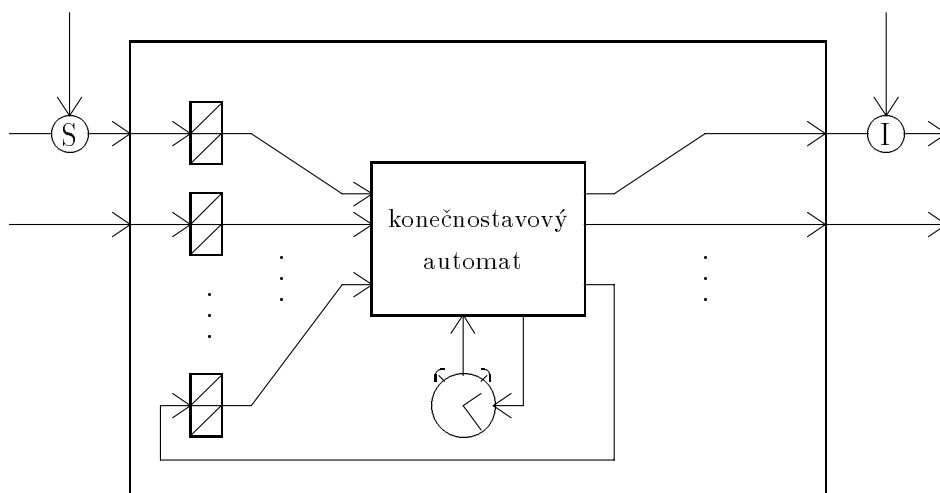


Obrázok 2: Základná štruktúra agenta.

2.2 Subsumpčná architektúra

V tejto práci je ako základ použitá už spomínaná **subsumpčná architektúra** radiča agentov. Spočíva v dekompozícii radiča na moduly podľa aktivity. Každý takýto modul predstavuje jedno správanie agenta. Výsledné správanie agenta je výsledkom interakcie týchto modulov.

Modul (obrázok 3) sa skladá z konečnostavového automatu, registrov a budíka. Do modulu vchádzajú vstupné signály od receptorov alebo iných modulov a uchováva sa v registroch. Na základe obsahu registrov konečnostavový automat vysiela výstupné signály do efektorov alebo iných modulov. Výstupný signál môže byť taktiež uchovaný v registri, čím vzniká možnosť zapamätania si nejakého údaja. Budík slúži na pozdržanie reakcie na vstupné signály alebo na rozloženie výstupných signálov do väčšieho časového úseku. Vstupný signál môže byť potlačený a nahradený signálom z iného modulu (supresia), čo umožňuje pripojiť na jeden vstup viac výstupov. Výstupný signál môže byť zase na základe signálu z iného modulu zastavený (inhibícia), čo umožňuje inému modulu na čas vyradiť modul z medzimodulovej interakcie. Supresia a inhibícia umožňujú kombinovať moduly do určitej hierarchie, kde nadradený modul spôsobuje supresiu alebo inhibíciu podradených modulov. Moduly potom možno radiť do vrstiev podľa funkcií ktoré zabezpečujú ([Brooks 86], [Brooks 89]). Z hľadiska technickej realizácie sa budík, supresia a inhibícia dajú zabezpečiť vhodnou manipuláciou s registrami, takže nemusia byť explicitnou súčasťou systému (nová subsumpcia). Na zaznamenanie konečnostavového automatu sa používa podoba špeciálneho algoritmu. Na jeho zápis bol vyvinutý špeciálny, tzv. behaviorálny jazyk postavený na báze jazyka Common Lisp ([Brooks 90]).



Obrázok 3

2.3 Architektúra stochastického agenta

V tejto práci navrhne a budeme využívať špecifickú formu subsumpčnej architektúry, ktorá bude

- čo najjednoduchšia
- úplne modulárna
- vhodná na simuláciu

a ktorá nebude priamo orientovaná na robotiku (neuvažuje asynchrónnosť, oneskorenie signálov a podobne).

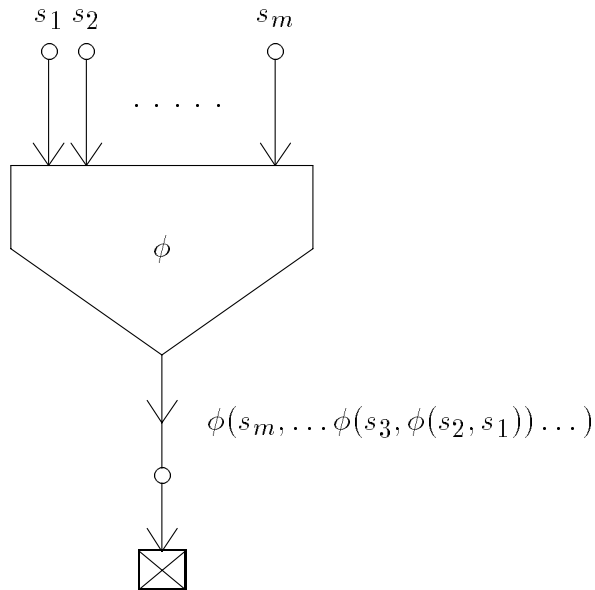
Prvá požiadavka vyplýva z nášho zámeru zaoberať sa možnosťami produkovania emergentného správania pri použití čo najjednoduchších agentov. Použití agenti budú čisto reaktívny, nebudú mať žiadnu pamäť a nebudú mať budík. Ich radič bude mať pravidelnú štruktúru, komunikácia medzi modulmi bude obmedzená na minimum.

Druhá požiadavka zabezpečuje, že pri zostavovaní agenta z modulov, môžeme kedykoľvek nejaký modul pridať alebo odobrať (pričom sa nemusíme starať o prepojenia medzi modulmi) bez následkov na funkčnosti celého systému (samozrejme s následkami na jeho správaní).

Tretia požiadavka vystihuje skutočnosť, že agenta nepotrebujeme stelesniť, ale potrebujeme ho simulovať. Pritom musíme prostredie agenta simulovať, čím drasticky obmedzíme rozsah jeho vstupov v porovnaní so stelesneným agentom. Túto „bohatosť“ vstupov budeme musieť niečím nahradiť.

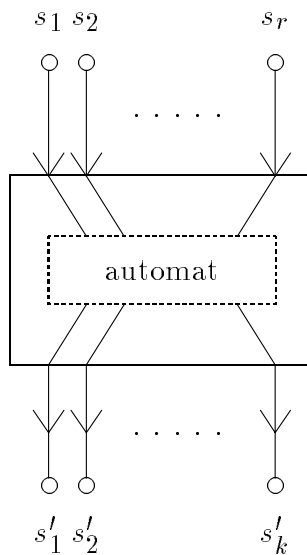
Agent tejto architektúry sa bude skladať z receptorov, efektorov a radiča. Radič sa bude skladať z dvoch úrovní modulov. V prvej vrstve budú moduly produkujúce jednotlivé správania (**behaviorálne moduly**), v druhej moduly zabezpečujúce interakciu (spoluprácu) modulov z prvej vrstvy (**koproductory**). Do behaviorálnych modulov budú vchádzať iba signály z receptorov. Na ich základe sú z behaviorálnych modulov vysielané signály do koproductorov, ktoré tieto signály spracúvajú a vysielajú na efektor. Pritom každému efektoru prislúcha práve jeden koproductor. Ten prijíma signály z viacerých (nie nutne všetkých) modulov, ktoré spracuje do jedného signálu a ten vyšle na svoj efektor. Spracovanie signálov koproductorom prebieha na základe binárnej asociatívnej a komutatívnej operácie nad signálmi (obrázok 4). Táto operácia je zvolená tak, aby niektoré signály mali vplyv na spracovanie iných signálov. Vyslaním takéhoto signálu môže behaviorálny modul spôsobiť napríklad inhibíciu signálov z iných modulov.

Tento spôsob medzimodulovej interakcie si môžeme predstaviť na známom príbehu o oslovi. Osol prechádza v strede medzi rovnako veľkými a rovnako chutnými kopami sena. Jeden modul preberajúci informácie z ľavého oka prikáže efektoru v nohách otočiť doľava. Ale modul spojený s pravým okom prikáže otočiť doprava. Koproductor majúci na starosti nohy to zloží a výsledkom je, že hladný osol prejde pomedzi kopy. (Neberte však tento príbeh ako spochybňovanie tejto architektúry, svet je menej dokonalý ako si myslíme.)



Obrázok 4: Koproduct so svojim efektorom.

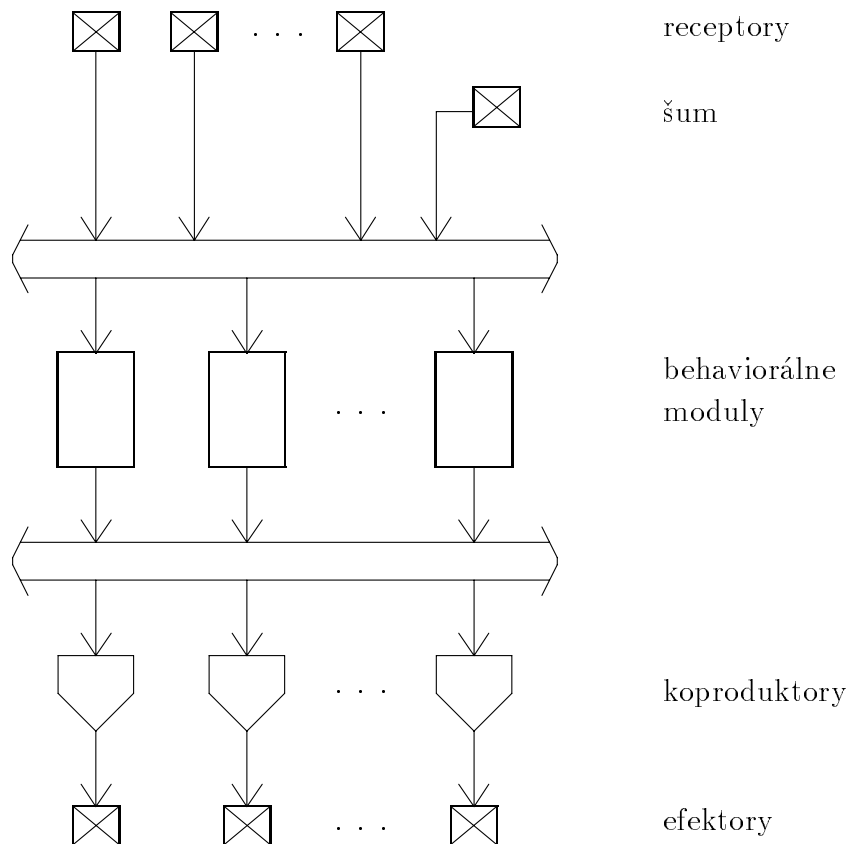
Už sme spomínali, že behaviorálne moduly prijímajú signály z receptorov a na ich základe vysielajú signály do koproduktorov. Teraz objasníme túto činnosť bližšie. Predpokladajme, že modul prijíma signály z r receptorov a vysielá signály na k koproduktorov. Behaviorálny modul obsahuje konečnostavový automat, ktorý každej vstupnej r -tici priradí jednu výstupnú k -ticiu. Automat by teda bolo možné zaznamenať ako zoznam dvojíc vstupov a výstupov. My ho však budeme zapisovať pomocou algoritmu v behaviorálnom jazyku.



Obrázok 5: Behaviorálny modul.

Už vieme, že podľa signálu z koproduktora efektor vykoná určitú akciu. Tá sa však nemusí vykonať úspešne, alebo môže mať vplyv na vlastnosti efektora. Aby sa o tom „dozvedeli“ behaviorálne moduly, máme k niektorým efektorom receptory indikujúce stav efektorov po vykonaní ich akcií.

Keď mechanická ruka uchopí predmet, ostane tento predmet uchopený. Keďže agent nemá možnosť zapamätať si, že predmet bol uchopený, musí byť k dispozícii niečo, čo vždy povie, či je v ruke predmet alebo nie.



Obrázok 6: Architektúra stochastického reaktívneho agenta.

Spomínali sme, že pri modelovaní systému dochádza k problémom, ktoré sú spôsobené nedostatkom podnetov v modelovanom prostredí agenta. Nie sme schopní namodelovať celý reálny svet, t.j. zabezpečiť, aby každé dve veci boli rôzne, i keď možno podobné. V modelovanom prostredí bude chýbať tzv. šum. Preto architektúru agenta obohatíme o receptory, ktoré budú vytvárať umelý šum. Signály z týchto „šumových“ receptorov budú pre agenta predstavovať šum v jeho prostredí, i keď budú v skutočnosti vznikať nezávisle od neho. Prítomnosť „šumových“ receptorov spôsobí, že niektoré konečnostavové automaty v behaviorálnych moduloch sa stanú nedeterministickými vzhľadom na tú časť vstupov, ktorá je prejavom skutočného vnímania modelovaného prostredia. Konkrétne to budú automaty tých modulos,

ktoré na vstup prijímajú aspoň jeden signál zo „šumového“ receptora. Určitému vstupu (nezahrňajúc signály zo „šumových“ receptorov) bude teda prislúchať viac výstupov, pričom každému výstupu bude prislúchať určitá pravdepodobnosť, súčet týchto pravdepodobností bude jednotka.

Keď pozorujeme červíka na doske stola, napriek tomu, že podľa nášho úsudku (a teda i schopností stôl modelovať) je doska všade rovnaká, červík buď nemá stále ten istý pohľad, alebo ak ho aj má, neprekáča mu to rozlične sa správať: napríklad nelezie po priamke (i keď niektoré väčšie úseky jeho cesty sú priame a stálo by zato zistiť, či je nejaká súvislosť medzi dĺžkou týchto úsekov a rôznymi atribútmi dosky). Keby sme modelovali túto situáciu, nemôžeme dať agentovi schopnosť vyčleňovať z tejto dosky rôzne objekty, lebo schopnosť vyčleňovať objekty si modelovaný agent požičiava od nás (od ľudí) a my schopnosť chápať dosku rôzne podľa miesta na nej nemáme. Jediné čo môžeme agentovi požičať, je generátor šumov, ktoré mu budú poskytovať podnety, ktoré mu nemôže poskytnúť nami namodelované prostredie. Následne na to sa vynára otázka, či to nie je podfuk, lebo vďaka variabilnosti správania za tých istých podmienok je agent schopný s mizivou ale nenulovou pravdepodobnosťou urobiť čokoľvek. My si však na jeho správaní budeme všímať iba to, čo urobí s pravdepodobnosťou blízkou jednotke, teda to, čo skutočne bude robiť. Navyše agent nemá žiadnu možnosť nedeterminizmus zneužívať, to máme iba my a my sa zaväzujeme, že to nebudeme robiť. Je to koniec koncov dosť namáhavé, lebo agent musí plniť akýsi cieľ a keď mu dáme možnosť výrazne robiť všetko možné, bude síce robiť všetko možné, ale nebude plniť stanovený cieľ. Ako mieru prípustného použitia nedeterminizmu uvedieme nasledujúci príklad:

Agent sa pohybuje priamo, nič sa okolo neho nenachádza, teda jeho vstupy sú stále rovnaké a keby bol deterministický, musel by tak ísť stále, lebo ako čisto reaktívny agent nemá vnútorný stav a teda ani netrpezlivosť. Modul, ktorý chodenie zabezpečuje, je však nedeterministický a s malou pravdepodobnosťou zmení smer pohybu, takže agent pôjde rovno (väčšinou) dlhú dobu, ale potom zatočí. Nedeterminizmus modulov teda dokáže čiastočne nahrádzať vnútorný stav agenta.

Tým sme presne špecifikovali architektúru agenta tak, ako ju budeme používať. Vzhľadom na nedeterminizmus behaviorálnych modulov a jeho pravdepodobnostnú povahu budeme takéhoto agenta nazývať **stochastickým reaktívnym agentom**. Architektúra stochastického reaktívneho agenta je znázornená na obrázku 6.

Skúsme sa teraz v krátkosti zamyslieť nad tým, nakoľko táto architektúra splňa naše očakávania:

O jej jednoduchosti nemôže byť pochyb. Má pravidelnú štruktúru. Komunikácia medzi jej časťami prebieha na veľmi nízkej úrovni, lebo behaviorálne moduly si neposielaajú žiadne správy a o činnosti iných modulov sa môžu dozvedieť len nepriamo - z jej následkov. Je tu uplatnený princíp „každý na vlastnú päť“ .

Úplná modularita systému je zaručená. Na špecifikovanie zloženia agenta nám stačí vymenovať, z ktorých behaviorálnych modulov sa skladá. Podľa toho je jasné aké receptory, koproduktory a efektory sú súčasťou agenta - sú to tie, na ktoré sa moduly odvolávajú. Keďže moduly medzi sebou priamo nekomunikujú, nie je potrebné starať sa o prepojenia medzi nimi. Následkom toho možno kedykoľvek modul zo systému odobrať alebo pridať bez straty funkčnosti.

O tom, že táto architektúra je vhodná na simuláciu, sa presvedčíme neskôr.

2.4 Komunikácia medzi agentami

Keď poznáme štruktúru agenta, môžeme stanoviť vzťahy medzi agentami. Kolóniu možno chápať ako súbor agentov, ale nemožno zabúdať na vzťahy medzi nimi, ktoré z nej robia čosi viac. Vo väčšine prác ([Parker 92b]) majú agenti nejaké identifikačné číslo a môžu si na základe toho posielat' (i na diaľku) správy. Podľa nás oveľa vhodnejší spôsob predstavuje **komunikácia** cez prostredie ([Mataric 92]). Agenti nebudú mať žiadne identifikačné čísla, takže pre jedného agenta budú ostatní agenti na nerozoznanie rovnakí. Rozdielne bude iba ich správanie sa v prostredí. Agenti môžu prostredie modifikovať tak, že to bude na iných agentov vplývať a tým bude prebiehať komunikácia. Takúto komunikáciu by už bolo vhodnejšie nazývať **interakciou**. Na jej uskutočnenie nestačí iniciatíva toho, kto interakciu spôsobuje, respektíve je potrebná aj iniciatíva toho kto jej bude podrobený. Je nemožné interakciu niekomu adresovať, ani dopredu zistiť na koho a kedy bude mať vplyv.

Kapitola 3

Modelovanie kolónie agentov

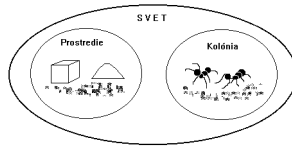
3.1 Kolónia ako súčasť sveta

Základný pojem pri modelovaní kolónie agentov bude pojem **objekt**. Objekt je každá entita, nachádzajúca sa v prostredí nejakého pozorovateľa, ktorý jej vie prisúdiť vlastnú individualitu (t.j. vie ju **vyčleniť z prostredia** a prehlásiť za jeden celok). Tento pozorovateľ navyše vie o dvoch objektoch pozorovaných v dvoch rôznych časových okamihoch povedať, či sú totožné, teda či sú tou istou entitou. Napriek tomu sa však dva totožné objekty môžu líšiť, pričom súbor znakov v ktorých sa môžu líšiť nazývame ich **stavom**. Pozorovateľ dokáže rozdeľovať objekty do skupín (**druhov**) na základe určitej analógie medzi nimi.

Keď vás pozorovateľ stretne na ceste k holičovi, vyčlení vás zo svojho prostredia a povie si: „Aha, toto je XY.“ Keď pôjdete ostrihaný naspäť, napriek tomu, že ste sa zmenili, povie si opäť: „Toto je XY“, lebo sa zmenil iba váš stav. Ale keď vás o chvíľu prejde parný valec, takže sa zmeníte po druhý raz, pozorovateľ si povie: „Toto možno bol(a), ale už nie je XY“. Keď vás pozorovateľ stretne s vašim kamarátom, povie si: „Hľa, obaja sú ľudia.“ Ale keď vás stretne s vašim psom, povie si: „Hľa, jeden človek a jeden pes“. Proces, akým to pozorovateľ robí, je dosiaľ neobjasnený, i keď túto schopnosť majú ľudia i zvieratá. Počítač na ktorom budeme kolóniu agentov modelovať, ju však nemá, takže sa pri modelovaní budeme musieť obmedziť na určitú skupinu objektov, ktoré budú tvoriť zjednodušený „svet“. V tejto skupine si urobíme poriadok takpovediac jednorázovým použitím našej vlastnej schopnosti vyčleňovať objekty z prostredia.

Z **objektov** budeme konštruovať **svet**, ktorého súčasťou bude kolónia agentov, ktorá nás zaujíma. Kolóniu teda nemožno oddeliť od prostredia, v ktorom sa nachádza. Konštrukcia bude založená na tom, že budeme **umiestňovať** objekty do nejakého priestoru. Pri výbere objektov, z ktorých budeme svet konštruovať, dodržíme podmienku, že budú „rovnako veľké“, čo znamená, že žiaden umiestňovaný objekt nebude zložením viacerých umiestňovaných objektov. **Priestor** sa tak rozpadne na množinu miest, ktoré sú také veľké, že do nich umiestnime práve jeden

umiestňovaný objekt. Časť umiestňovaných objektov budú objekty predstavujúce **agentov**. Skupina týchto objektov bude predstavovať **kolóniu**. Ostatné objekty budú predstavovať **prostredie**, v ktorom sa kolónia nachádza (pozri obrázok 7). Musíme si dať samozrejme pozor na to, aby sme naraz do priestoru neumiestnili dva totožné objekty, lebo to odporuje našim predstavám. Ďalej budeme pre zjednodušenie predpokladať, že na jedno miesto možno umiestniť len jeden objekt. Umiestnením objektov do priestoru dostaneme „strnulý“ svet, ktorý budeme nazývať **momentkou**.



obrázok 7

Nás samozrejme zaujíma, ako sa bude kolónia správať a o tom sa dá hovoriť iba vzhľadom na určité plynutie času. **Čas** budeme chápať ako nekonečnú postupnosť časových okamihov, pričom budeme predpokladať, že svet v nasledujúcom časovom okamihu možno dostať uplatnením určitých pravidiel vývoja na svet v aktuálnom časovom okamihu.

Vývoj kolónie bude pre nás teda postupnosť momentiek, kde každá nasledujúca momentka vzniká z aktuálnej aplikovaním spomínaných pravidiel vývoja. Prvú momentku v tejto postupnosti budeme nazývať počiatkovou momentkou kolónie.

Budeme predpokladať, že všetky zmeny sveta spôsobujú len agenti a že prostredie sa z vlastnej iniciatívy meniť nedokáže. Agenti menia prostredie na základe voľby a následného vykonávania akýchsi akcií. **Pravidlá vývoja** si potom možno predstaviť ako orákulum, ktoré si vypýta od agentov akcie ktoré sa rozhodli vykonať a na základe toho zmení čo treba. Mohlo by sa zdať, že orákulum má ľahkú úlohu a že mu stačí postupne vykonať všetky dané akcie, ale nie je to tak. Orákulum musí zabezpečiť paralelné vykonanie všetkých akcií, tj. vykonať ich naraz. Pritom vznikajú konfliktné situácie, ktoré musí vedieť rozpoznať a „spravodlivo“ riešiť.

Keď chcú napríklad dvaja agenti vstúpiť na to isté voľné miesto, orákulum musí rozhodnúť, kto z nich tam vstúpi a kto narazí na úspešnejšieho kolegu. Aby bolo orákulum spravodlivé, musí si v takomto prípade hodiť mincou. Ešte zložitejšia situácia vzniká, keď sa agent rozhodne vstúpiť na miesto, na ktorom už nejaký agent stojí. Keby sa mu striktno povedalo že narazil, mohlo by sa stať, že miesto bude nakoniec voľné (agent z neho odíde) a agent napriek tomu narazil. To je absurdné a preto musí orákulum tieto prípady zvlášť zabezpečiť. Ešte zložitejší prípad vzniká, keď sa chcú agenti pohybovať „do kruhu“. Vtedy už naozaj nemožno nahradiť paralelnú hromadnú akciu postupnosťou akcií, lebo žiadna z nich sa jednotlivito nedá vykonať, ale spolu sa vykonať dajú.

Keďže orákulum bude vlastne v konfliktných prípadoch zabezpečovať „spoluprácu“ medzi agentami, budeme ho nazývať i **koprodukčná funkcia**.

3.2 Výber akcií agenta

V predchádzajúcej podkapitole sme sa zaoberali modelovaním kolónie. Predpokladali sme, že v každom okamihu sa pre každého agenta vie, aké akcie sa rozhodol vykonať. Teraz sa budeme zaoberať spôsobom, akým sa tieto akcie vyberajú. Budeme teda popisovať vzťah medzi jednotlivými časťami agenta, jeho umiestnením, orientáciou a aktuálnou momentkou.

Na základe aktuálnej momentky, umiestnenia a orientácie agenta budú receptory agenta vnímať prostredie v ktorom sa agent nachádza a tieto vnemy budú premieňať na signály. Výnimkou v tomto prípade budú „šumové“ receptory, ktoré budú generovať náhodný signál. Tieto signály budú receptory vysielat' do behaviorálnych modulov.

Behaviorálne moduly stanovia na základe vstupných signálov z receptorov výstupné signály, ktoré budú vyslané na koprodukory. Ich stanovenie prebieha na základe práce konečnostavového automatu. Každý modul tým navrhne vykonať určité akcie.

Koprodukory signály z behaviorálnych modulov zložia do jediného signálu a ten vyšlú na svoj efektor. Zloženie prebieha pomocou binárnej operácie, ktorá je asociatívna a komutatívna, takže nezáleží na poradí spracovávaných signálov. Táto operácia je definovaná spravidla tak, aby sa modulom dala možnosť vyslaním určitého signálu čiastočne alebo úplne potlačiť vplyv signálov z iných modulov. Takisto môže umožniť „zloženie“ signálov, pokiaľ ide o akciu, ktorá sa môže vykonať s rôznou intenzitou (napr. otočenie). Týmto spôsobom prebieha výber akcií z návrhov behaviorálnych modulov. Niekedy bude ako výsledok prijatý návrh jedného modulu, inokedy to bude kompromis medzi návrhmi viacerých modulov. Výsledkom tohto „dohadovacieho konania“ budú akcie, ktoré sa skutočne budú vykonávať.

Efektory podľa signálov zo svojich koproduktorov už len vykonajú príslušné akcie, ktoré týmto signálom zodpovedajú.

Kapitola 4

Matematický model kolónie agentov

4.1 Vzťah agentov a prostredia v čase

- **Priestor** je nejaká neprázdna množina P . Jej prvky nazývame **miesta**.
- Množina **smerov** v priestore P je neprázdna konečná množina S , ku ktorej existuje funkcia *posun* : $P \times S \rightarrow P$.
- Množina **druhov** je určitá neprázdna konečná množina D .
- Množina **objektov** je množina O bližšie nešpecifikovaných konštruktov, ku ktorej
 - (1) existuje relácia $\equiv \subseteq O^2$ taká, že pre každé $o_1, o_2 \in O$ je $o_1 \equiv o_2$ práve vtedy, keď o_1 je ten istý objekt ako o_2 , len v inom stave.
 - (2) existuje funkcia *druh* : $O \rightarrow D$ taká, že pre každé $o_1, o_2 \in O$ je $druh(o_1) = druh(o_2)$ práve vtedy, keď o_1 je toho istého druhu ako o_2 . (Platí $\forall o_1, o_2 \in O : o_1 \equiv o_2 \implies druh(o_1) = druh(o_2)$.)
 - (3) existuje funkcia *miesto* : $O \rightarrow P$, ktorá dáva miesto, na ktorom sa objekt $o \in O$ nachádza.
 - (4) existuje funkcia *smer* : $O \rightarrow S$, ktorá dáva smer, v ktorom je objekt $o \in O$ orientovaný.

Táto množina je zjednotením dvoch disjunktných množín: množiny agentov \mathcal{A} a množiny prostredia \mathcal{P} .

- **Momentka** je taká podmnožina $s \subseteq O$, že platí

$$\forall o_1, o_2 \in s : o_1 \not\equiv o_2 \\ \text{miesto}(o_1) \neq \text{miesto}(o_2)$$

Množinu všetkých momentiek označíme symbolom \mathcal{S} .

- Množina **akcií** je konečná množina C , ku ktorej existuje **koprodukčná funkcia** $\coprod : \mathcal{S} \times 2^C \rightarrow \mathcal{S}$, pričom $\coprod(s, C')$ je momentka, ktorá vznikne z momentky s súčasným aplikovaním akcií C' .
- Množina **agentov** je podmnožina množiny objektov $\mathcal{A} \subseteq O$ taká, že existuje funkcia *akcia* : $\mathcal{A} \times \mathcal{S} \rightarrow 2^C$, ktorá dáva množinu akcií, pre vykonanie ktorých sa určitý agent, nachádzajúci sa v danej momentke, rozhodol.
- **Vývoj kolónie** je postupnosť momentiek $\{s_i\}_{i=0}^{\infty}$, kde

$$s_{i+1} = \coprod\left(s_i, \bigcup_{a \in \mathcal{A} \cap s_i} C'(a, s_i)\right)$$

a $C'(a, s_i) \in \text{akcia}(a, s_i)$. Momentku s_0 nazývame **počiatočnou momentkou kolónie**.

4.2 Vzťah agenta a jeho akcií v jednom okamihu

- **Konštanty** sú prvky dostatočne veľkej, ale konečnej podmnožiny nejakej číselnej množiny Z . Slúžia ako ekvivalenty signálom, ktoré receptory vysielajú do behaviorálnych modulov, ktoré tieto moduly vysielajú do koproduktorov a ktoré koproduktory vysielajú svojim efektorom.
- **Šumy** sú prvky nekonečnej, ale ohraničenej číselnej množiny S . Sú to ekvivalenty signálov, ktoré vysielajú „šumové“ receptory.
- **Receptor** je funkcia $r_i : \mathcal{S} \times P \times S \rightarrow Z \cup \{S\}$ taká, že obrazom jej oboru je buď Z alebo $\{S\}$ (podľa toho, či je efektor „šumový“ alebo nie). Množinu všetkých receptorov označíme $\mathcal{R} = \{r_1, r_2, \dots, r_{pr}\}$.
- **Efektor** je funkcia $v_i : Z \rightarrow C$. Množinu všetkých efektorov označíme $\mathcal{E} = \{e_1, e_2, \dots, e_{pe}\}$.
- **Koproduktor** je binárna operácia na množine konštánt $\phi_i : Z^2 \rightarrow Z$. ϕ_i slúži na vyhodnotenie signálu, ktorý bude poslaný na efektor e_i . Množinu všetkých koproduktorov označíme $\Phi = \{\phi_1, \phi_2, \dots, \phi_{pk}\}$. Platí, že $pk = pe$.
- **Behaviorálny modul** je usporiadaná trojica

$$m_i = \left(\underbrace{\{k_i^1, k_i^2, \dots, k_i^{p_i}\}}_{K_i}, \underbrace{\{l_i^1, l_i^2, \dots, l_i^{q_i}\}}_{L_i}, f_i \right)$$

kde k_j sú indexy receptorov od ktorých modul signály prijíma, l_j sú indexy koproduktorov na ktoré modul signály vysiela a

$$f_i : \prod_{K_i} Z \cup S \longrightarrow \prod_{L_i} Z$$

je funkcia, ktorá každej p_i -tici signálov, ktoré prídu z receptorov priraďuje q_i -tici signálov, ktoré modul vyšle na koprodukory. Množinu všetkých modulov označíme $\mathcal{M} = \{m_1, m_2, \dots, m_{pm}\}$.

Keď budeme za vstup do modulu považovať len tie signály z receptorov, ktoré vznikajú na základe skutočného vnímania prostredia (teda tie, ktoré neprichádzajú zo „šumových“ receptorov), každému takto zúženému vstupu do modulu bude prislúchať viac výstupov. Každému výstupu bude potom prislúchať určitá pravdepodobnosť úmerná počtu nezúžených vstupov, ktoré tento vstup zahŕňa. Tento stav môžeme presnejšie popísať reláciou

$$g_i \subseteq \prod_{K'_i} Z \times \prod_{L_i} Z \times (0, 1)$$

kde $K'_i \subseteq K_i$ sú indexy receptorov, ktoré nie sú „šumové“. Táto relácia popisuje, pre aký výstup w a s akou pravdepodobnosťou p dáva modul pre zúžený vstup v' . Definovaná je nasledovne: $g(v', w, p)$ práve vtedy, keď

(1) existuje vstup v taký, že $f_i(v) = w$ a $pr_{K'_i}(v) = v'$ (keď zúžime vstup v dostaneme v')

$$(2) p = \frac{|\{v \mid f_i(v) = w, pr_{K'_i}(v) = v'\}|}{|\{v \mid pr_{K'_i}(v) = v'\}|}$$

Pritom samozrejme platí, že pre každé v' bude $\sum_{g(v', w, p)} p = 1$.

- K množine \mathcal{A} existujú nasledujúce funkcie:

$$receptory : \mathcal{A} \longrightarrow 2^{\mathcal{R}}$$

$$efektory : \mathcal{A} \longrightarrow 2^{\mathcal{E}}$$

$$moduly : \mathcal{A} \longrightarrow 2^{\mathcal{M}}$$

$$koproduktory : \mathcal{A} \longrightarrow 2^{\Phi}$$

pre ktoré platí, že pre každé $a_1, a_2 \in \mathcal{A}$ je

$$receptory(a_1) = receptory(a_2)$$

$$a_1 \equiv a_2 \implies moduly(a_1) = moduly(a_2) \iff druh(a_1) = druh(a_2)$$

$$efektory(a_1) = efektory(a_2)$$

$$koproduktory(a_1) = koproduktory(a_2)$$

- K množine agentov existuje funkcia

$$\text{stav-efektorov} : \mathcal{A} \longrightarrow \{st : \text{efektory}(a) \longrightarrow Z^{|\text{efektory}(a)|}\}$$

ktorá každému agentovi priradí funkciu, ktorá o každom jeho efektore hovorí, či akcia, ktorá sa mala vykonať v predchádzajúcom okamihu, bola vykonaná úspešne alebo nie (t.j. či bola akcia vykonaná koprodukčnou funkciou \coprod alebo nie). Funkcia *stav-efektorov* je potrebná nato, aby mohli receptory, indikujúce úspešnosť vykonania akcií, zistiť potrebnú skutočnosť z aktuálnej momentky. Pre receptor r indikujúci stav na efektore e_j bude potom platiť:

$$r(s, m, d) = \text{stav-efektorov}(a)(e_j)$$

kde a je agent nachádzajúci sa v momentke s na mieste m .

- Hodnota funkcie $\text{akcia} : \mathcal{A} \times \mathcal{S} \longrightarrow 2^C$ pre agent a a momentku s sa na základe vnútornej štruktúry agenta stanoví nasledovne:

- (1) pre každý z receptorov $r_i \in \text{receptory}(a)$ sa vyhodnotí signál $\varrho(r_i)$, ktorý bude poslaný do modulu:

$$\begin{aligned} \varrho(r_i) &= r_i(s) && \text{ak } r_i \text{ nie je „šumový“} \\ \varrho(r_i) &\in r_i(s) && \text{ak } r_i \text{ je „šumový“} \end{aligned}$$

- (2) pre každý behaviorálny modul $m_i \in \text{moduly}(a)$ sa na základe týchto signálov ϱ určia signály μ_{m_i} , ktoré budú vyslané na koproduktory:

$$\mu(m_i) = f_i\left(\varrho(r_{k_i^1}), \varrho(r_{k_i^2}), \dots, \varrho(r_{k_i^{p_i}})\right)$$

- (3) v každom koproduktore $\phi_i \in \text{koproduktory}(a)$ sa signály, ktoré prišli od behaviorálnych modulov (nie nutne od všetkých) zložia pomocou binárnej operácie ϕ_i do jediného signálu $\kappa(\phi_i)$:

$$\kappa(\phi_i) = \phi_i\{ \mu(m_j) \mid m_j \in \text{moduly}(a), \exists x : k_j^x = i \}$$

- (4) na základe týchto signálov sa pre každý efektor $e_i \in \text{efektory}(a)$ stanoví akcia $c(e_i) \in C$, ktorú bude tento efektor následne vykonávať:

$$c(e_i) = v_i(\kappa(\phi_i))$$

- (5) akcie $\text{akcia}(a, s)$, ktoré bude agent následne vykonávať sa potom stanovia ako zjednotenie akcií jeho jednotlivých efektorov:

$$\text{akcia}(a, s) = \{ c(e_i) \mid e_i \in \text{efektory}(a) \}$$

Kapitola 5

Kritérium kompetencie

5.1 Špecifikácia experimentu

Pri experimente budeme sledovať modelovaný vývoj nejakej kolónie, pre ktorý sú špecifikované tri nasledujúce pojmy ([Goodwin 93]):

- (1) **kolónia**, t.j. akí agenti tvoria kolóniu a koľko ich je
- (2) **prostredie**, t.j. v akom prostredí sa agenti nachádzajú
- (3) **úloha**, t.j. aký spoločný cieľ majú agenti dosiahnuť

Na jednu časť experimentu (sledovanie jedného vývoja) sa môžeme pozeráť ako na overovanie riešenia nasledujúceho problému:

Je dané prostredie, úloha (cieľ) a schopnosti agentov (mechanizmus). Aké aktivity (radič) a akú početnosť majú mať jednotlivé druhy agentov, aby kolónia z nich zložená vyriešila úlohu (dosiahla cieľ)?

Riešením takto definovaného problému je špecifikácia jednotlivých druhov agentov a prípadne i stanovenie ich početnosti.

Na celý experiment sa potom môžeme pozeráť ako na riešenie nasledujúceho problému:

Je daný charakter prostredia, teda druhy objektov tvoriacich prostredie a podmienky kladené na ich umiestnenie. Ďalej je daná úloha a schopnosti agentov. Aké aktivity a pomerné početnosti majú mať jednotlivé druhy agentov, aby v určitom rozsahu konkrétnych prostredí existovala ku každému prostrediu kolónia zložená zo špecifikovaných druhov agentov (s početnosťami závislými od prostredia), ktorá rieši úlohu (dosahuje cieľ)?

Riešením takto stanoveného problému je špecifikovanie jednotlivých druhov agentov a hľadanie intervalov v početnostiach objektov prostredia a týchto druhov agentov, v ktorých dochádza k riešeniu úlohy (dosiahnutiu cieľa). My sa nebudeme zaoberať presnými hranicami týchto intervalov, tie vo väčšine prípadov neexistujú, ale budeme pozorovať zmeny spôsobené zvyšovaním alebo znižovaním početností jednotlivých druhov objektov z určitého vybraného stavu. Výber tohto stavu bude prebiehať na základe intuície a meraných kvantitatívnych ukazovateľov tak, aby pri tomto stave prostredia a kolónie vznikali emergentné efekty. Kolónie budeme teda skúmať **experimentálne**³.

5.2 Kritérium riešenia úlohy

Hovorili sme už o tom, že kolónia rieši úlohu alebo dosahuje cieľ. Zatiaľ sme však bližšie neurčili kedy dochádza k tomuto javu. Naša predstava je taká, že k vágnej formulácii úlohy (cieľa) existuje nejaká **vyhodnocovacia funkcia** $e : \mathcal{S} \rightarrow \mathcal{C}$, ktorá pre ľubovoľnú momentku číselne vyhodnotí nakoľko je úloha splnená ([Goodwin 93]). Koobor tejto funkcie je nejaká číselná množina \mathcal{C} , volí sa pre konkrétnu úlohu a záleží od jej povahy. V tejto práci sme si zvolili, že úloha je tým viac vyriešená, čím menšia je hodnota vyhodnocovacej funkcie. K úlohe ďalej existuje určitá medza $\nu \in \mathcal{C}$. Poklesnutie hodnoty vyhodnocovacej funkcie pod túto medzu signalizuje vyriešenie úlohy (dosiahnutie cieľa).

Nech $\{s_i\}_{i=0}^{\infty}$ je vývoj kolónie v prostredí. Ďalej nech je daná úloha s vyhodnocovacou funkciou e a medzou ν . Keď bude existovať j také, že pre všetky $i \geq j$ bude $e(s_i) \leq \nu$, budeme hovoriť, že **kolónia rieši úlohu v danom prostredí**.

5.3 Kritérium kompetentného riešenia úlohy

Kritérium riešenia úlohy hovorí len o tom, či kolónia rieši alebo nerieši úlohu, ale nič nehovorí o tom, ako ju rieši. Je jasné, že nie všetky kolónie, ktorých činnosť spôsobuje vyriešenie úlohy, spĺňajú intuitívnu predstavu o skutočnom riešení. Pri kolónii nás bude zaujímať hlavne to, či existuje kvalitatívny rozdiel medzi aktivitami jedného osamoteného agenta a kolónie, ktorá sa z takýchto agentov skladá.

Nech je úlohou napíliť drevo na zimu a máme na to dvoch ľudí. Keď bude píliť každý osve svojou pílkou, napíliť drevo dvakrát rýchlejšie ako jeden. Ale keď budú píliť spolu jednou pílkou, pričom ju každý bude držať na opačnom konci, môže sa stať, že drevo napíliť až trikrát rýchlejšie (obrázok 8). Vysvetlenie tohto javu možno hľadať v tom, že v dreve sú uzly, ktoré píliacich veľmi zdržujú. Ak sa sila ktorou sa bude píliť zdvojnásobí, rýchlosť prekonania uzlov nemusí vzrásť len dvojnásobne, lebo vzťah medzi píliacou silou a rýchlosťou prekonania uzla nemusí byť (a ani nie je) lineárny.

³ I keď v súlade s teoretickým výskumom kolónií v [Kelemen - Kelemenová 92].

Vidíme, že prvom prípade nastala použitím väčšieho počtu ľudí iba kvantitatívna, ale nie kvalitatívna zmena ich práce. V druhom prípade však došlo použitím väčšieho počtu ľudí ku kvalitatívnej zmene, čo sa nakoniec prejavilo aj v neočakávane veľkej kvantitatívnej zmene.



Obrázok 8: Nekompetentné a kompetentné riešenie úlohy

Budeme teda hovoriť o kompetentnom a nekompetentnom riešení úlohy, pričom v prvom prípade budeme mať na mysli aktivity, ktoré nielen spôsobia vyriešenie úlohy, ale ktoré navyše úlohu vyriešia spôsobom, ktorému intuitívne prisudzujeme prívlastok „rozumný“.

O **kompetencii** budeme hovoriť len v prípade konkrétneho stanovenia kolónie (i s početnosťami jednotlivých druhov) v konkrétnom prostredí. Pritom ak nebude výrazne záležať na konkrétnom umiestnení objektov, budeme za konkrétne prostredie považovať iba špecifikáciu početností jednotlivých druhov prostredia.

Na zavedenie pojmu kompetencie budeme potrebovať pojem **kolónie polovičnej veľkosti**, skrátene $1/2$ -kolónia. Zavedenie tohto pojmu vychádza z predpokladu, že ku každej kolónii existuje $1/2$ -kolónia, ktorá je jej funkčnou polovicou. V prípade, že sa kolónia skladá z agentov jedného druhu, bude to kolónia s polovičným počtom agentov. V prípade viacdruhovej kolónie sa početnosti jednotlivých druhov budú meniť v závislosti na funkcii, ktorú v kolónii plnia, najčastejšie sa budú redukovať na polovicu alebo budú ostávať nezmenené.

Nech je teda daná kolónia, jej $1/2$ -kolónia, prostredie a úloha. Budeme hovoriť, že kolónia rieši úlohu v tomto prostredí kompetentne, ak v tomto prostredí úlohu rieši a navyše ju rieši viac než dvakrát lepšie ako jej $1/2$ -kolónia. Čo však znamená lepšie riešiť úlohu? To záleží na povahe úlohy. V našich experimentoch bude mierou kvality riešenia čas, za ktorý poklesne hodnota vyhodnocovacej funkcie pod určitú hodnotu. Rozdiel dvojnásobku tohto času pre $1/2$ -kolóniu a analogického času pre kolóniu pre určitú hodnotu z kooboru vyhodnocovacej funkcie potom udáva, či je pre túto hodnotu kolónia viac než dvakrát lepšia ako $1/2$ -kolónia (> 0) alebo nie (< 0). Pokiaľ je tento rozdiel počnúc istou hodnotou až do hodnoty indikujúcej vyriešenie úlohy stále kladný, bude kolónia riešiť úlohu viac než dvakrát lepšie ako $1/2$ -kolónia.

Priebeh spomínaného rozdielu pre všetky hodnoty z obrazu vyhodnocovacej funkcie bude krivka, ktorú budeme nazývať krivka kompetencie. Tvar tejto krivky transparentnejšie znázorňuje, či kolónia rieši úlohu kompetentne, alebo nie. Dôležitá je hlavne tá časť krivky, ktorá prislúcha okoliu medze vyriešenia úlohy. Pokiaľ je

krivka na určitom okolí medze celá nad x -ovou osou, kolónia rieši úlohu kompetentne. Naopak, ak takéto okolie neexistuje, kolónia rieši úlohu nekompetentne.

Prejdime teraz k exaktným definíciám. Nech je daná kolónia, prostredie a úloha. Nech $e : \mathcal{S} \rightarrow \mathcal{C}$ je vyhodnocujúcou funkciou tejto úlohy a nech $\nu \in \mathcal{C}$ je jej medza.

Na základe toho definujeme funkciu $f_1 : \{0, 1, \dots, \infty\} \rightarrow \mathcal{C}$ ako priemernú hodnotu vyhodnocovacej funkcie pre všetky možné vývoje kolónie v danom prostredí. Takže

$$f_1(t) = \frac{\sum_{\{s_i\}_{i=0}^{\infty} \in \mathcal{V}} e(st)}{|\mathcal{V}|}$$

kde \mathcal{V} je množina všetkých možných vývojov kolónie v danom prostredí. Podobne definujeme funkciu $f_2 : \{0, 1, \dots, \infty\} \rightarrow \mathcal{C}$ ako priemernú hodnotu vyhodnocovacej funkcie pre $1/2$ -kolóniu v tom istom prostredí.

Krivkou kompetencie budeme nazývať funkciu $cc : \mathcal{C} \rightarrow \{0, 1, \dots, \infty\}$, pre ktorú platí:

$$cc(h) = f_2^{-1}(h) - 2 \cdot f_1^{-1}(h)$$

kde $f_i^{-1}(h) = \min\{t \mid f_i(t) \leq h\}$.

Kritérium kompetencie bude nasledovné: Ak existuje okolie medze úlohy $\langle h_0, \nu \rangle$ také, že pre všetky h také, že $h_0 \leq h \leq \nu$, je $cc(h) > 0$, budeme hovoriť, že kolónia rieši úlohu v danom prostredí kompetentne. V opačnom prípade budeme hovoriť, že kolónia rieši úlohu v danom prostredí nekompetentne.

5.4 Ohraničenia spôsobené experimentálnym prístupom

Pri experimentoch dochádza z dôvodu absencie absolútnych údajov k určitým ohraničeniam, ktoré následne uvedieme.

Ohraničenia sa v prvom rade týkajú stanovenia funkcií f_i . Nie je samozrejme možné zmerať vyhodnocujúce funkcie pre všetky možné vývoje. Preto sa uspokojíme s určitým počtom vývojov, pre ktoré budeme vyhodnocujúcu funkciu merať a funkcie f_i stanovovať ako priemer nameraných hodnôt. Tento spôsob práce podporí skutočnosť, že výsledky jednotlivých meraní sa nebudú príliš líšiť.

Ďalej si musíme dať pozor na určovanie kompetencie. Okolie medze, v ktorom je krivka kompetencie kladná bude musieť byť dostatočne veľké, aby sme boli schopní zodpovedne prehlásiť, že to nie je náhoda spôsobená výberom vývojov. Z toho istého dôvodu je nebezpečný výraz $cc(h) > 0$ v kritériu kompetencie, v praxi ho musíme nahradiť výrazom $cc(h) \gg 0$. Avšak ako sa ukáže neskôr, nebudú s určovaním kompetencie pri experimentoch žiadne problémy.

Kapitola 6

Charakterizácia experimentov

Na vykonávanie experimentov bol vyvinutý programový produkt **Mra**. Pri jeho realizácii bola snaha urobiť čo najotvorenejší systém, ale predsa sa ukázalo, že je potrebné pristúpiť k špecifickejšiemu definovaniu niektorých častí systému.

6.1 Priestor

Priestorom je v systéme **Mra** štvorčeková sieť obdĺžnikových rozmerov, Y riadkov po X stĺpcoch⁴. Každý štvorček (resp. políčko) bude predstavovať jedno **miesto**. Máme teda priestor

$$P_{\mathbf{Mra}} = \{ (y, x) \mid 0 \leq y \leq Y + 1, 0 \leq x \leq X + 1 \}$$

V skutočnosti má priestor $Y + 2$ riadkov a $X + 2$ stĺpcov. Tieto „prebytočné“ miesta slúžia na umiestnenie takých objektov prostredia, ktoré zabránia agentom odísť von z priestoru.

6.2 Smery a pohyb

Pohyb agentov v takomto priestore možno založiť buď na štvor- alebo osem- príľahlosti. Vzhľadom na predstavu stelesnenia agentov vyhovuje lepšie prvá možnosť.

⁴ Y a X sú v systéme obmedzené vzhľadom na rozsah pamäti, ktorú poskytuje použitá technika. Pri všetkých experimentoch bol použitý rozmer 64x47 ani nie preto, že to uľahčovalo grafické zobrazenie, ale preto, že pri maximálnom počte objektov $\doteq 3000$ ostala asi tretina pamäte **heap** voľná, čo umožnilo experimenty vykonávať bez rizika ich násilného a nečakaného prerušenia.

Máme teda smery

$$S_{\text{Mra}} = \{(-1, 0), (0, -1), (1, 0), (0, 1)\}$$

pričom funkcia *pohyb* : $P_{\text{Mra}} \times S_{\text{Mra}} \rightarrow P_{\text{Mra}}$ určujúca pohyb je definovaná nasledovne:

$$\text{pohyb}((y, x), (d_y, d_x)) = \begin{cases} (y + d_y, x + d_x) & \text{pre } 1 \leq y \leq Y, 1 \leq x \leq X \\ (y, x) & \text{inak} \end{cases}$$

Predpokladá, že miesta $(0, x)$, $(Y + 1, x)$, $(y, 0)$, $(y, X + 1)$ sú nedosiahnuteľné.

Pod bezprostredným okolím agenta umiestneného na (y, x) budeme rozumieť miesta $(y - 1, x)$, $(y, x - 1)$, $(y + 1, x)$ a $(y, x + 1)$.

Lesným mravcom by bolo iste veľmi nepríjemné, keby ich niekto nútil pohybovať sa podľa vyššie uvedenej pohybovej funkcie. Agentom to bude jedno. Dôležitejšia než spôsob pohybu je jeho samotná existencia. V prírode existuje mnoho spôsobov pohybu, každý však slúži tomu istému účelu.

6.3 Koprodukčná funkcia

Koprodukčná funkcia \coprod_{Mra} plní dve úlohy:

- stanovuje usporiadanie akcií vykonávaných v jednom okamihu
- „spravodlivo“ rieši konfliktné situácie

Prvá úloha je potrebná preto, lebo agent naraz vyberá na vykonanie viacero akcií, pričom niektoré tieto akcie majú z hľadiska okamihu vykonania rozdielne postavenie. Toto postavenie však vyplýva z ich podstaty a nie je ho ťažké ani stanoviť, ani zabezpečiť.

Napríklad sa agent v jednom okamihu rozhodne otočiť sa doľava a ísť. Sú to dve akcie vykonávané v jedinom okamihu. Napriek tomu je jasné, že sa má agent najprv otočiť doľava a až potom ísť. Je to dané tým, že otočenie sa viaže k danému okamihu, zatiaľ čo chodenie sa viaže k plynutiu času medzi daným okamihom a nasledujúcim.

Podstata zabezpečenia druhej úlohy sa zakladá na rozdelení akcií na tie, ktoré menia umiestnenie agenta a na ostatné. Najprv sa vykonávajú tie druhé, potom prvé.

Na dosiahnutie „spravodlivosti“ stačí u tých druhých vykonávať akcie v náhodnom poradí. Konfliktná situácia tu totiž spočíva iba v tom, že z viacero akcií možno vykonať iba jednu. Konfliktná situácia je lokálna a jej vplyv sa neprenáša na iné miesta priestoru. Preto stačí náhodne vybrať jednu z akcií.

Zložitejšie to bude v prvom prípade. Tu sa totiž konflikty prenášajú v priestore. Pokiaľ sa chce viac agentov premiestniť na jedno miesto, výber toho, ktorý to skutočne bude, ovplyvní agentov, ktorí sa chcú premiestniť na ich miesta, atď. Pri zvolení poradia akcií, treba preto najprv náhodne (a „spravodlivo“) určiť, čie akcie

budú vykonávané v konfliktných prípadoch a potom treba vždy uprednostňovať tie akcie, ktoré možno vykonať (ich vykonaním sa akcie dovtedy nevykonateľné môžu stať vykonateľnými). Ešte ostáva vyriešiť konflikt spočívajúci v pohybe „do kruhu“, keď máme reťaz agentov, kde sa nasledujúci chce premiestniť na miesto predchádzajúceho, zacyklená. Vtedy treba takéto cykly posunúť, i keď to nie je možné vykonávaním jednotlivých posunov v nejakom rozumnom poradí. Preto na chvíľu jedného agenta z cyklickej reťaze odstránime z momentky a po vykonaní posunových akcií ostatných agentov z reťaze ho doplníme na správne miesto.

Koprodukčná funkcia musí ďalej zabezpečiť, aby stavy efektorov (hlavne tých, ktorých stav vníma nejaký receptor) zodpovedali tomu, či akcia nimi predložená koprodukčnej funkcii na vykonanie, vykonaná bola alebo nie, respektíve, či zmenila stav efektora alebo nie.

6.4 Druhy objektov

Objekty z modelu kolónie sú v programe `Mra` reprezentované technológiou objektovo orientovaného programovania. Použitý je triedno-inštančný prístup. Druh objektu je potom trieda objektov a D_{Mra} je množina všetkých tried. S takouto definíciou druhov sa dá dobre porovnávať, či majú dva objekty rovnaký druh, ale horšie sa už robia zložitejšie operácie. Preto má každý druh svoje identifikačné číslo. Navyše je v `Mra` implementovaná funkcia, ktorá z objektu vyčíta identifikačné číslo jeho druhu. Vďaka tomu sa na celú vec možno pozeráť tak, že máme k dispozícii injekciu $\xi : D_{\text{Mra}} \rightarrow Z$, ktorá druhu (t.j. triede objektov) priraduje jeho identifikačné číslo. (Toto číslo potom možno chápať aj ako signál). Na označenie faktu, že sa niekde objekt nenachádza, budeme používať konštantu `nothing` $\in Z$, ktorá neslúži ako identifikačné číslo žiadnemu druhu: `nothing` $\notin \xi [D_{\text{Mra}}]$.

Základné druhy objektov prostredia (ďalšie pribúdajú pri jednotlivých experimentoch) sú kameň a stena. Oba majú podobnú povahu, ale slúžia na iné účely. Kamene predstavujú základný druh prekážky. Steny slúžia na vyplnenie miest $(0, x)$, $(Y+1, x)$, $(y, 0)$, $(y, X+1)$, čím tieto miesta urobia nedosiadateľnými. Na schopnosti agentov sa pritom kladie požiadavka, aby nedokázali steny ničiť alebo premiestňovať. Na označenie identifikačných čísel druhov kameň a stena budeme používať konštanty `stone`, `wall` $\in Z$.

Druhy agentov budú špecifikované až pre konkrétny experiment ako súbory receptorov, efektorov, behaviorálnych modulov a koproduktorov (pričom na ich jednoznačné určenie stačí uviesť behaviorálne moduly).

6.5 Šumy

Ako ekvivalenty šumom, teda signálom od „šumových“ receptorov, sú v **Mra** použité čísla z intervalu $\langle 0, 1 \rangle$.

6.6 Konštanty

Ako ekvivalenty ostatných signálov používa program **Mra** celé čísla, presnejšie ich bližšie nešpecifikovanú ale konečnú podmnožinu⁵. Nie je potrebné sa starať o ich presný číselný rozsah.

6.7 Receptory

Základné druhy receptorov⁶ v programe **Mra** (ďalšie možno samozrejme dedefinovať) vnímajú, čo sa nachádza v bezprostrednom okolí agenta v určitom smere. Sú teda štyri druhy receptorov:

Straight :

$$(s, (y, x), (d_y, d_x)) \mapsto \begin{cases} \xi(\text{druh}(o)) & \exists o \in s : \text{miesto}(o) = (y + d_y, x + d_x) \\ \text{nothing} & \text{inak} \end{cases}$$

Left :

$$(s, (y, x), (d_y, d_x)) \mapsto \begin{cases} \xi(\text{druh}(o)) & \exists o \in s : \text{miesto}(o) = (y - d_x, x + d_y) \\ \text{nothing} & \text{inak} \end{cases}$$

Right :

$$(s, (y, x), (d_y, d_x)) \mapsto \begin{cases} \xi(\text{druh}(o)) & \exists o \in s : \text{miesto}(o) = (y + d_x, x - d_y) \\ \text{nothing} & \text{inak} \end{cases}$$

Back :

$$(s, (y, x), (d_y, d_x)) \mapsto \begin{cases} \xi(\text{druh}(o)) & \exists o \in s : \text{miesto}(o) = (y - d_y, x - d_x) \\ \text{nothing} & \text{inak} \end{cases}$$

Ďalej máme k dispozícii „šumové“ receptory typu

Rnd :

$$(s, (y, x), (d_y, d_x)) \mapsto \{\langle 0, 1 \rangle\}$$

⁵ Názvy konštánt budú zložené zo symbolov **a, b, ... , z, !**.

⁶ Na rozdiel od konštánt budú názvy častí agenta zložené zo symbolov **a, b, ... , z, A, B, ... , Z, 0, 1, ... , 9**, pričom budú začínať veľkým písmenom.

6.8 Koproductory, efektory a akcie

Základné druhy efektorov umožňujú pohyb agenta. Ďalšie druhy zabezpečujúce uchopenie a polozenie predmetov, likvidovanie druhých objektov a podobne, budú dodefinované.

Pohyb umožňuje efektor `Move`, ktorý na základe signálu z koproduktora vykoná akciu, ktorá agenta premiestni na miesto $(x + d_x, y + d_y)$ alebo nevykoná žiadnu akciu, takže agent ostane na mieste. Koproduktor tohto efektoru dokáže prijať od behaviorálnych modulov tri druhy signálov: $\text{go}, \text{stop}, \text{go!} \in Z$. Tieto signály skladá do jedného na základe binárnej operácie ϕ_{Move} , ktorá je definovaná v tabuľke 1. Pokiaľ koproduktor vyšle na efektor signál `go!` alebo `go`, tak efektor vykoná akciu, následkom ktorej sa agent premiestni (presnejšie: dá ju vykonať koprodukčnej funkcii). Ak vyšle signál `Stop` efektor nevykonáva, žiadnu akciu a agent sa nepresunie.

ϕ_{Move}	go	stop	go!
go	go	stop	go!
stop	stop	stop	go!
go!	go!	go!	go!

Tabuľka 1

Riadenie pohybu, teda otáčanie, umožňuje efektor `Turn`, ktorý na základe signálu z koproduktora vykoná akciu, ktorá agenta pootočí o $0^\circ, 90^\circ, 180^\circ$ alebo 270° . Koproduktor tohto efektoru dokáže prijať od behaviorálnych modulov osem druhov signálov: $\text{no!}, \text{no}, \text{left}, \text{back}, \text{right}, \text{left!}, \text{back!}, \text{right!} \in Z$.

ϕ_{Turn}	no!	no	left	back	right	left!	back!	right!
no!	no!	no!	no!	no!	no!	no!	no!	no!
no	no!	no	left	back	right	left!	back!	right!
left	no!	left	back	right	no	left!	back!	right!
back	no!	back	right	no	left	left!	back!	right!
right	no!	right	no	left	back	left!	back!	right!
left!	no!	left!	left!	left!	left!	left!	no!	no!
back!	no!	back!	back!	back!	back!	no!	back!	no!
right!	no!	right!	right!	right!	right!	no!	no!	right!

Tabuľka 2

Tieto signály skladá do jedného na základe binárnej operácie ϕ_{Turn} , ktorá je definovaná v tabuľke 2. Pokiaľ koproduktor vyšle na efektor signál **no!** alebo **no**, tak efektor nevykoná žiadnu akciu a ak naň vyšle jeden zo signálov **left** alebo **left!**, **back** alebo **back!**, **right** alebo **right!** tak efektor vykonáva akciu, ktorej následkom je pootočenie agenta o 90° , 180° , prípadne 270° proti smeru hodinových ručičiek.

Vzhľadom na to, že k efektoru jednoznačne prináleží jeho koproduktor, budeme často hovoriť len „efektor“ namiesto „efektor a jeho koproduktor“.

6.9 Receptory indikujúce stav na efektoroch

Kým akcie efektora **Turn** možno vykonať vždy, akcie efektora **Move** nemusia byť vždy vykonateľné. Pri snahe vykonať akciu, ktorá agenta presunie na iné miesto, môže byť na tomto mieste objekt prostredia alebo iný agent, alebo sa taktiež môže stať, že je síce voľné, ale koprodukčná funkcia naň presunie iného agenta. Vtedy dochádza k stavu na efektore **Move**, ktorý zodpovedá nárazu. Na indikáciu nárazu slúži receptor **MoveStatus**, ktorý vysiela dva druhy signálov: **collision**, **ok**. Prvý bude signalizovať, že akcia spôsobujúca posun agenta nebola (nemohla byť) vykonaná, druhý opak, t.j. že táto akcia bola vykonaná alebo že na vykonanie nebola vybratá.

MoveStatus :

$$r(s, m, d) = \text{stav-efektorov}(a)(\text{Move}) = \begin{cases} \text{collision} \\ \text{ok} \end{cases}$$

kde $a \in s$ a $\text{miesto}(a) = m$.

6.10 Behaviorálne moduly

Základné behaviorálne moduly zabezpečujú pohyb a vyhýbanie sa prekážkam. Ďalšie moduly možno dodefinovať.

Na definovanie modulu je potrebné uviesť, z ktorých receptorov do modulu prichádzajú signály, na aké efektor (koproduktor) signály vysiela a prechodovú funkciu jeho konečnosťového automatu. Z praktických dôvodov budeme prechodovú funkciu zapisovať podmieneným pravidlom. V podmienkach tohoto pravidla sa budeme názvami receptorov odvolávať na signály, ktoré z nich vychádzajú. V akčnej časti pravidiel budeme výrazmi typu **Efektor(signál)** stanovovať aký signál sa má vyslať na aký koproduktor (ten je daný svojim efektorom). V každej akčnej časti podmieneného pravidla budeme uvádzať takéto výrazy pre všetky koproduktory s ktorými je behaviorálny modul spojený a oddeľovať ich budeme bodkočiarkami.

Modul **Moving** zabezpečuje neustály nepokoj agenta sprevádzaný pohybom a to nie stále priamočiarym. Modul zabezpečuje, aby sa agent nepohyboval stále po tej istej uzavretej trajektórii.

Modul **Avoidance** zabezpečuje vyhýbanie sa prekážkam.

Moving

```
receptory:      Rnd
efektory:       Move, Turn
telo:          if Rnd < 1/40
                then Turn(left); Move(go)
                else Turn(no); Move(go)
```

Avoidance

```
receptory:      Straight, Rnd
efektory:       Turn
telo:          if Straight = nothing
                then Turn(no)
                else
                  if Rnd < 1/2
                    then Turn(left)
                    else Turn(right)
```

Kapitola 7

Experiment: Dezaktivácia územia

7.1 Prostredie

Prostredie obsahuje dva druhy objektov: kamene a zamorené územia. Na označenie identifikačných čísel ich druhov budeme používať konštanty `stone` a `infestarea`. Tieto objekty sú v počiatočnej momentke náhodne umiestnené v priestore. Rozmery priestoru sú 64×47 miest.

7.2 Schopnosti agentov

Agenti sú schopní vnímať svoje bezprostredné okolie a rozpoznávať v ňom kamene a zamorené územia. Ďalej sú schopní pohybovať sa smere svojej orientácie, otáčať sa a dezaktivovať zamorené územia, nachádzajúce sa v ich bezprostrednom okolí v smere ich orientácie. Agenti nedokážu premiestňovať alebo ničiť kamene.

7.3 Úloha

Úlohou kolónie agentov je dezaktivovať čo najviac zamorených území.

Vyhodnocujúcou funkciou tejto úlohy je funkcia $e : \mathcal{S} \rightarrow N$, ktorá priradí momentke počet zamorených miest, teda:

$$e(s) = |\{ o \in s \mid \xi(\text{druh}(o)) = \text{infestarea} \}|$$

Všimnime si, že e je na momentkách ľubovoľného vývoja nerastúca.

Medzou tejto úlohy je malé prirodzené číslo, zvolíme $\nu = 10$. Tu si všimnime, že keby sme zvolili za medzu nulu, mohlo by sa s dosť veľkou pravdepodobnosťou stať, že kolónia nemá šancu úlohu vyriešiť. Môže sa totiž stať, že na všetkých miestach v bezprostrednom okolí nejakého zamoreného miesta sa nachádzajú kamene. Vtedy agenti nemajú možnosť toto zamorené miesto dezaktivovať. Úlohu budeme teda považovať za vyriešenú, keď počet zamorených miest klesne na 10 alebo pod 10.

7.4 Riešenie: zloženie agenta a kolónie

Kolónia navrhnutá na riešenie bude obsahovať agentov jediného druhu. Agent bude pozostávať:

- z receptorov `Straight`, `Left`, `Right`, `MoveStatus`, `Rnd`
- z efektorov `Move`, `Turn`, `CleanUp` a
- z behaviorálnych modulov `Moving`, `Avoidance` a `CleaningUp`.

Pomocou efektora `CleanUp` agent dokáže dezaktivovať zamorené územie, nachádzajúce sa v jeho bezprostrednom okolí v smere jeho orientácie. Koproduktor tohto efektora pracuje so signálmi `cleanup`, `no` a `no!`, ktoré skladá na základe binárnej operácie ϕ_{CleanUp} , ktorá je definovaná v tabuľke 3.

ϕ_{CleanUp}	no!	cleanup	no!
no!	no!	no!	no!
cleanup	no!	cleanup	cleanup
no	no!	cleanup	no

Tabuľka 3

Modul `CleaningUp` zabezpečuje dezaktiváciu zamoreného územia, ktoré sa ocitne v bezprostrednom okolí agenta. Modul spôsobí takú zmenu orientácie agenta, aby sa zamorené územie nachádzalo v smere jeho orientácie a následne spôsobí dezaktiváciu použitím efektora `CleanUp`. Aby mohol túto činnosť nerušene uskutočniť, spôsobuje supresiu signálov od ostatných modulov, ktoré prichádzajú na koprodukory efektorov `Move` a `Turn`.

`CleaningUp`

receptory: `Straight`, `Left`, `Right`

efektory: `CleanUp`, `Turn`, `Move`

```

telo:      if Straight = infestarea
           then CleanUp(cleanup); Turn(no!); Move(stop)
           else
             if Left = infestarea
               then CleanUp(no); Turn(left!); Move(stop)
             else
               if Right = infestarea
                 then CleanUp(no); Turn(right!); Move(stop)

```

Ostatné časti agenta sú základné a sú popísané v kapitole 6.

7.5 Výsledky modelovania vývoja kolónie

So zvolenou kolóniou sme pomocou systému *Mra* uskutočnili viacero modelovaní jej vývoja vo zvolenom prostredí, pričom sme merali hodnoty vyhodnocovacej funkcie. Výsledky potvrdili, že umiestnenie objektov v tomto prostredí na ne nemá veľký vplyv. Preto na konkretizáciu prostredia stačí uvádzať počty kameňov a zamorených území. Na konkretizáciu kolónie stačí uviesť počet agentov. Počiatočná momentka sa potom vytvorí na základe náhodného umiestnenia všetkých objektov.

Náš postup bol nasledovný. Urobili sme najprv niekoľko (≈ 20) modelovaní kolónie pre rôzne hodnoty početností kameňov, zamorených území a agentov. Nerobili sme žiadne merania, iba sme sledovali, čo sa deje a snažili sa vysvetliť pozorované javy, ktoré často budili dojem, že ich produkujú „živé tvory“. Na základe toho sme dospeli k nasledujúcim pozorovaniam, ktoré vyjadrujú vzťah medzi počiatočnou momentkou a priebehom modelovania:

- (1) Čím je menej agentov, tým menej je zaujímavá kvalita ich práce. Pre malý počet agentov (1, 2) existujú miesta na ktoré sa nikdy nedostanú. Pomerne rýchlo dezaktivujú zamorené územia vo svojom okolí, ale meniť svoj operačný priestor je pre ne obtiažne. Po určitom čase (≈ 2500) začne rýchlosť dezaktivácie rapídne klesať. Každý agent má tendenciu pohybovať sa po určitej triede trajektórií, ktorá je určená jeho počiatočným umiestnením a umiestnením objektov prostredia, hlavne kameňov.
- (2) Príliš veľa agentov dezaktivuje všetky zamorené územia (až na tie, ktoré sú nedostupné) príliš rýchlo na to, aby sa dalo čokoľvek spoľahlivo zmerať. V nameraných výsledkoch by boli veľké výkyvy, lebo hodnota vyhodnocujúcej funkcie klesá veľkými skokmi.
- (3) Čím je menej kameňov, tým menšia je trieda trajektórií po ktorej sa agent pohybuje. To spôsobuje pokles počtu stretnutí agentov a tým pádom aj interakcie medzi nimi. V konečnom dôsledku to spôsobuje aj pokles rýchlosti dezaktivácie.

- (4) Čím viac je kameňov, tým je trieda trajektórií po ktorých sa agent pohybuje bohatšia. Stúpa i intenzita stretnutí agentov, avšak od určitého počtu kameňov začne klesať ich vplyv na správanie agentov. To je spôsobené tým, že rýchlosť dezaktivácie sa zrýchli viac, takže agenti dezaktivujú všetky zamorené územia skôr, ako sa stihne prejaviť vplyv interakcie. Pri zvyšovaní počtu kameňov však existuje určitá medza, po prekročení ktorej začne veľkosť triedy trajektórií po ktorých sa agent pohybuje prudko klesať, lebo kamene začnú vytvárať súvislé steny, čo pri určitom počte kameňov znemožní meniť operačný priestor. Navyše pri veľkom počte kameňov sa priestor rozpadne na uzavreté časti, čím sa úloha stane neriešiteľnou.
- (5) Čím menej je zamorených území, tým sa pozorovanie stáva ťažším a výsledky náhodnejšími. To nie je nič prekvapujúce, keďže úloha je definovaná na základe ich počtu.
- (6) Čím viac je zamorených území, tým väčšia je trieda trajektórií, po ktorých sa agent pohybuje. To má tie isté dôsledky ako väčší počet kameňov.

Na základe týchto pozorovaní sme vytipovali nasledujúcu konfiguráciu početností, pri ktorej má interakcia agentov vplyv na ich správanie a pri ktorej je i možné túto interakciu spoľahlivo merať: 200-300 kameňov, 300-400 zamorených území, 8-30 agentov. Prostredie sme preto zostavili z 250-tich kameňov a 350-tich zamorených území a kolóniu z 20-tich agentov, takže jej $1/2$ -kolónia pozostávala z 10-tich agentov.

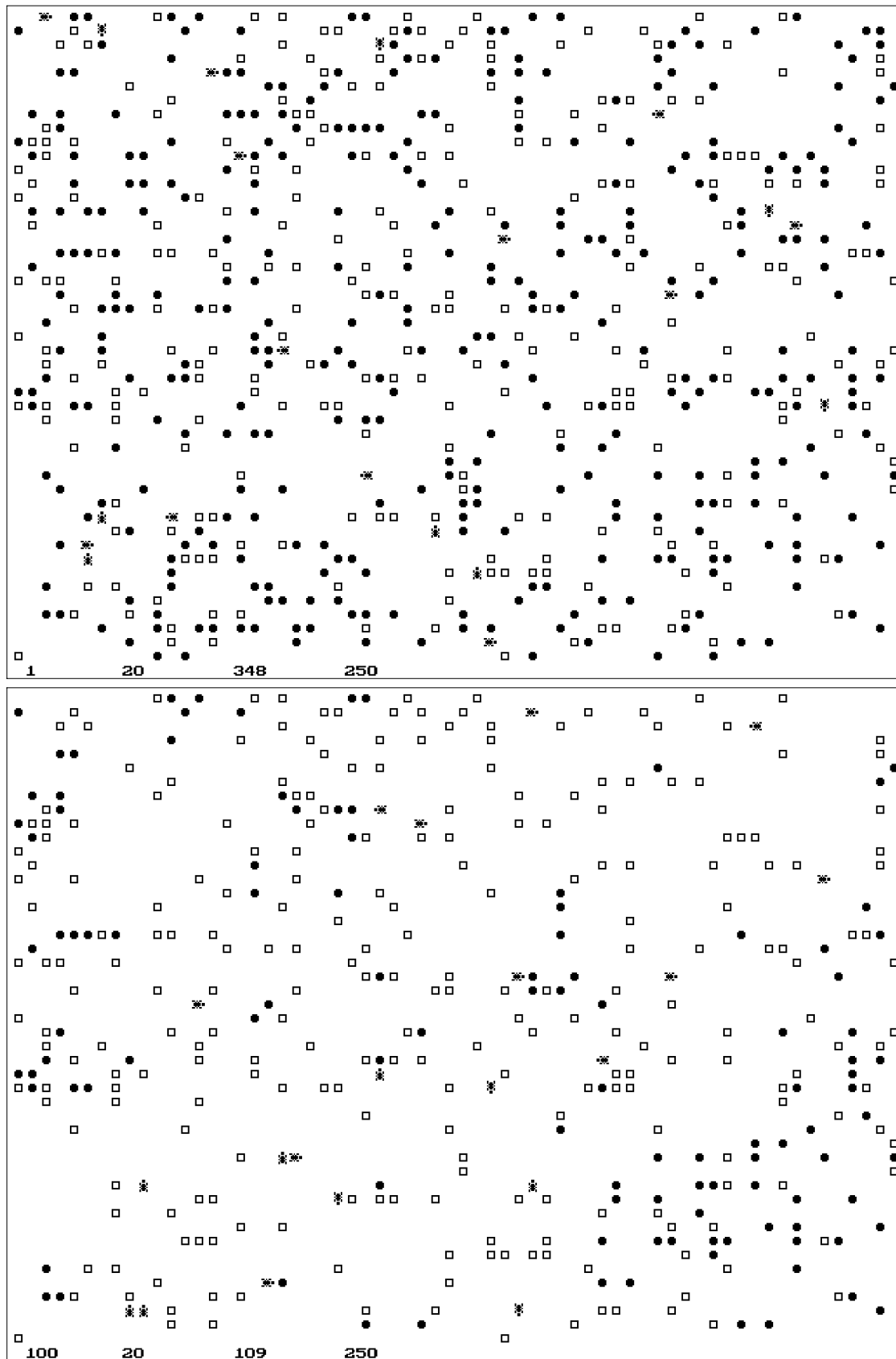
Pre takto zvolenú kolóniu a prostredie sme uskutočnili desať modelovaní pre rôzne umiestnenia objektov v počiatočnej momentke, ktoré sprevádzali merania hodnôt vyhodnocovacej funkcie⁷. Modelovania samozrejme zachytávali vývoj iba do určitého okamihu. Na základe predchádzajúcich modelovaní sme modelovali prvých 2000 okamihov vývoja. Jedno modelovanie trvalo zhruba 20 minút (použitá technika: PC 386DX, 40MHz). Nie je prekvapujúce, že takmer všetok tento čas bol spotrebovaný na vykonávanie koprodukčnej funkcie. Namerané údaje (hodnoty vyhodnocovacej funkcie a okamihy v ktorých boli namerané) predstavovali 160kB. Z týchto desiatich meraní sme urobili priemer a tento sme použili ako funkciu f_1 .

Podobne sme uskutočnili desať meraných modelovaní pre $1/2$ -kolóniu⁸ a priemer z nich sme použili ako funkciu f_2 .

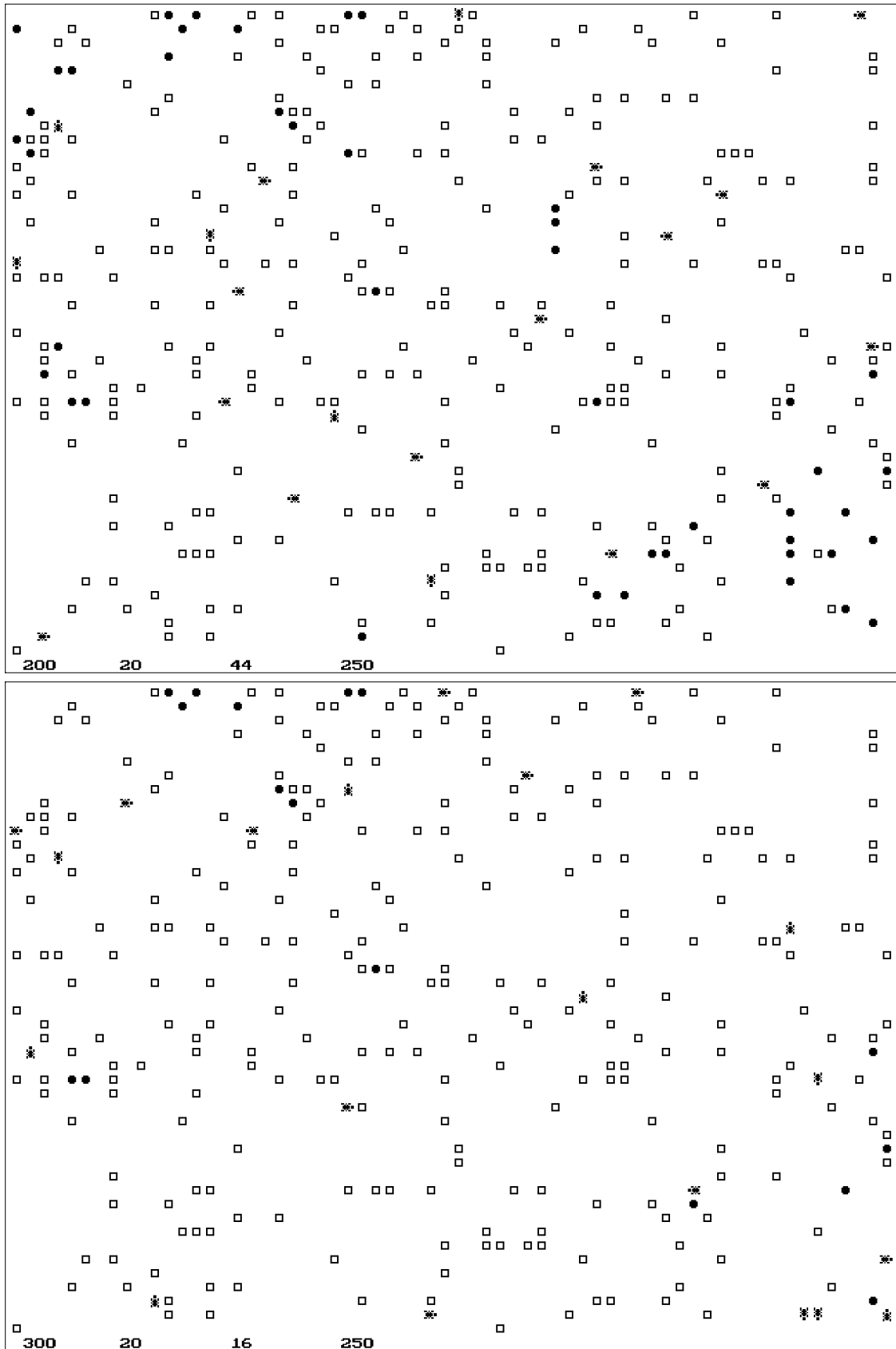
Vývoj kolónie môžeme sledovať na obrázkoch 9a-9h. Tri čísla v spodnom riadku obrázkov predstavujú aktuálny okamih a aktuálny počet agentov, zamorených území a kameňov. Vývoj $1/2$ -kolónie môžeme zasa sledovať na obrázkoch 10a-10l.

⁷ Uskutočniť modelovanie s meraním je náročnejšie ako uskutočniť modelovanie bez merania.

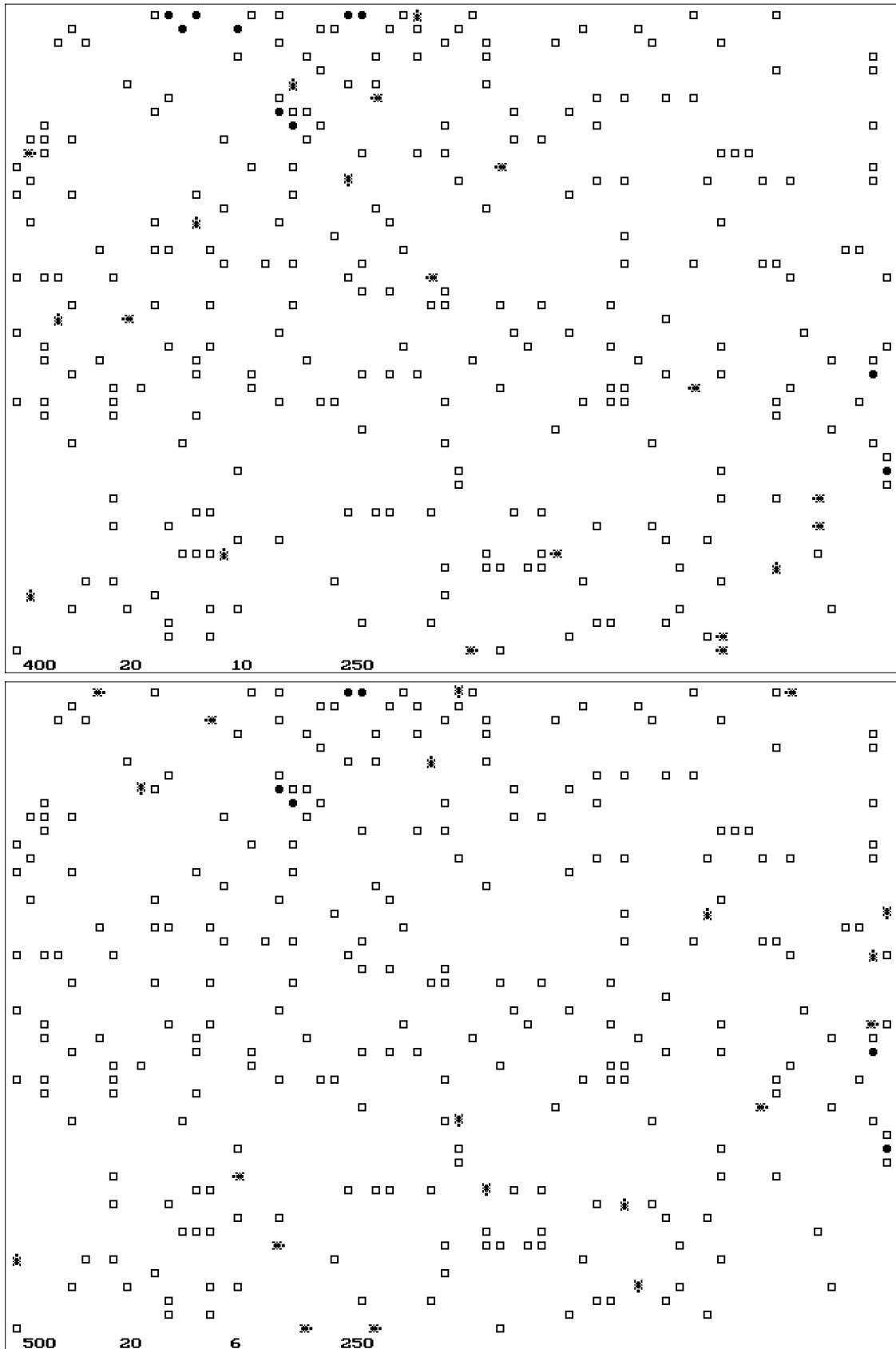
⁸ Modelovanie pre polovičný počet agentov trvalo približne rovnaký čas ako v predchádzajúcom prípade, čo svedčí o tom, že v algoritme realizujúcom koprodukčnú funkciu možno urobiť výrazné zlepšenia.



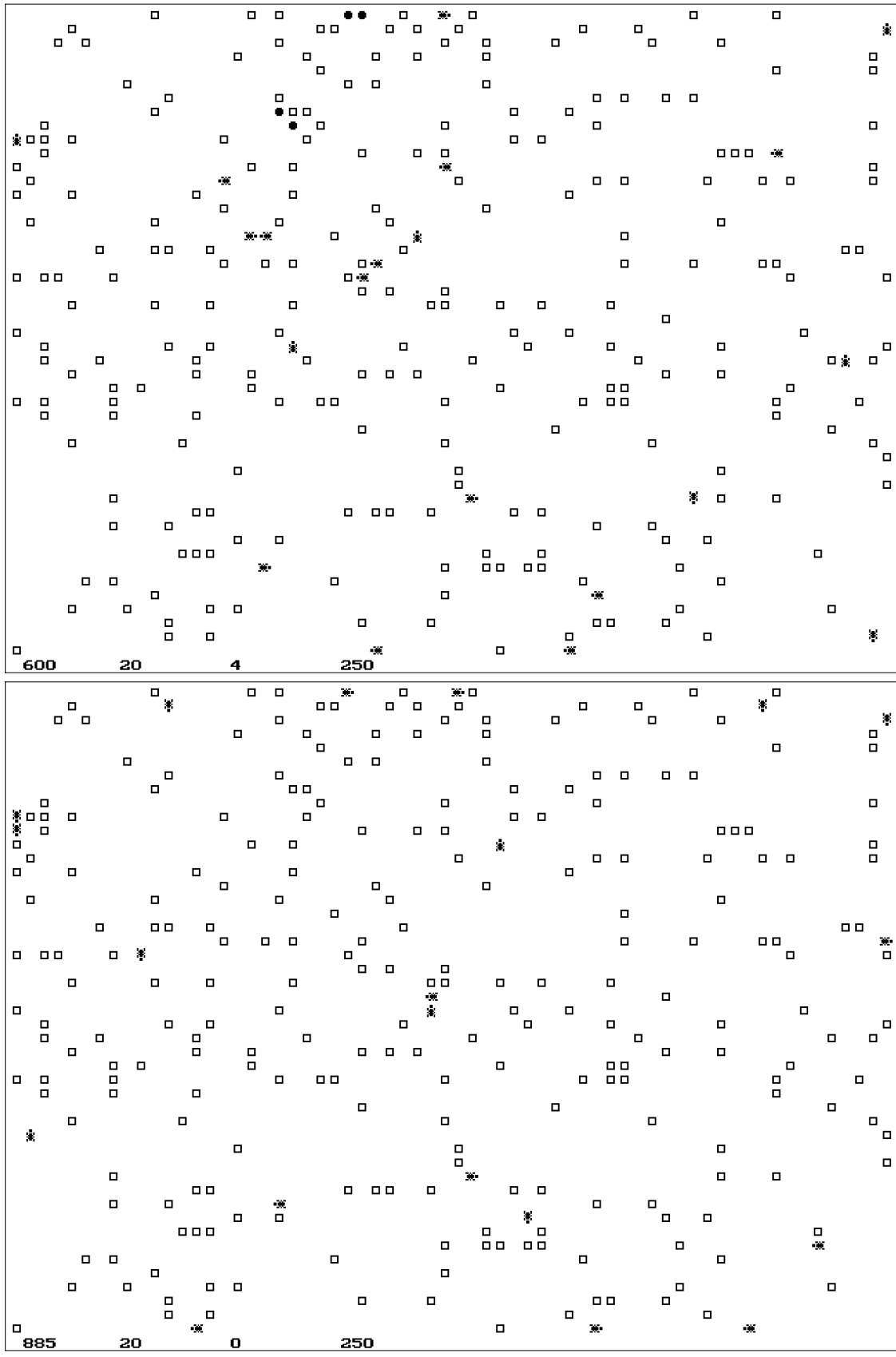
Obrázky 9a, 9b



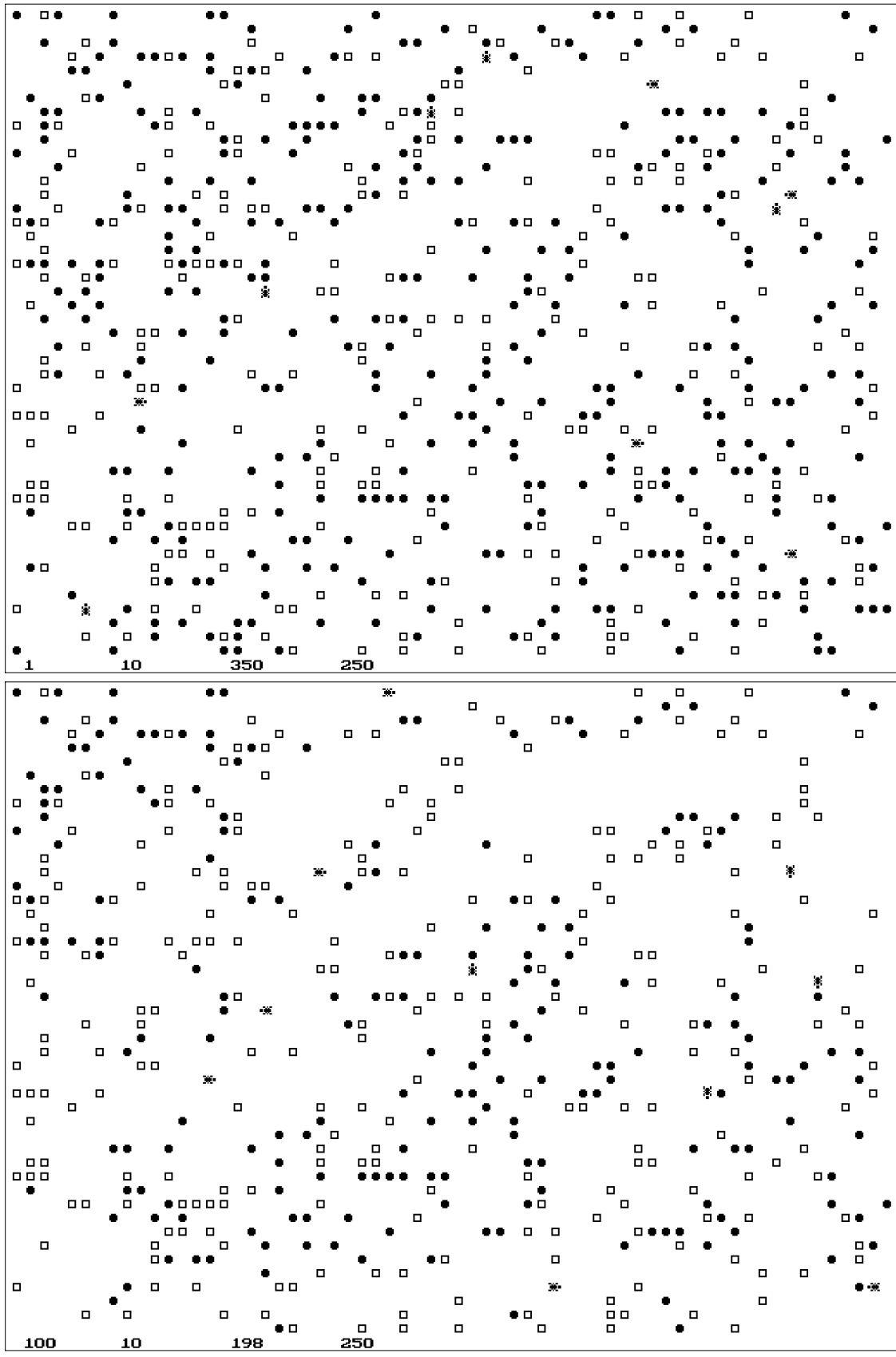
Obrázky 9c, 9d



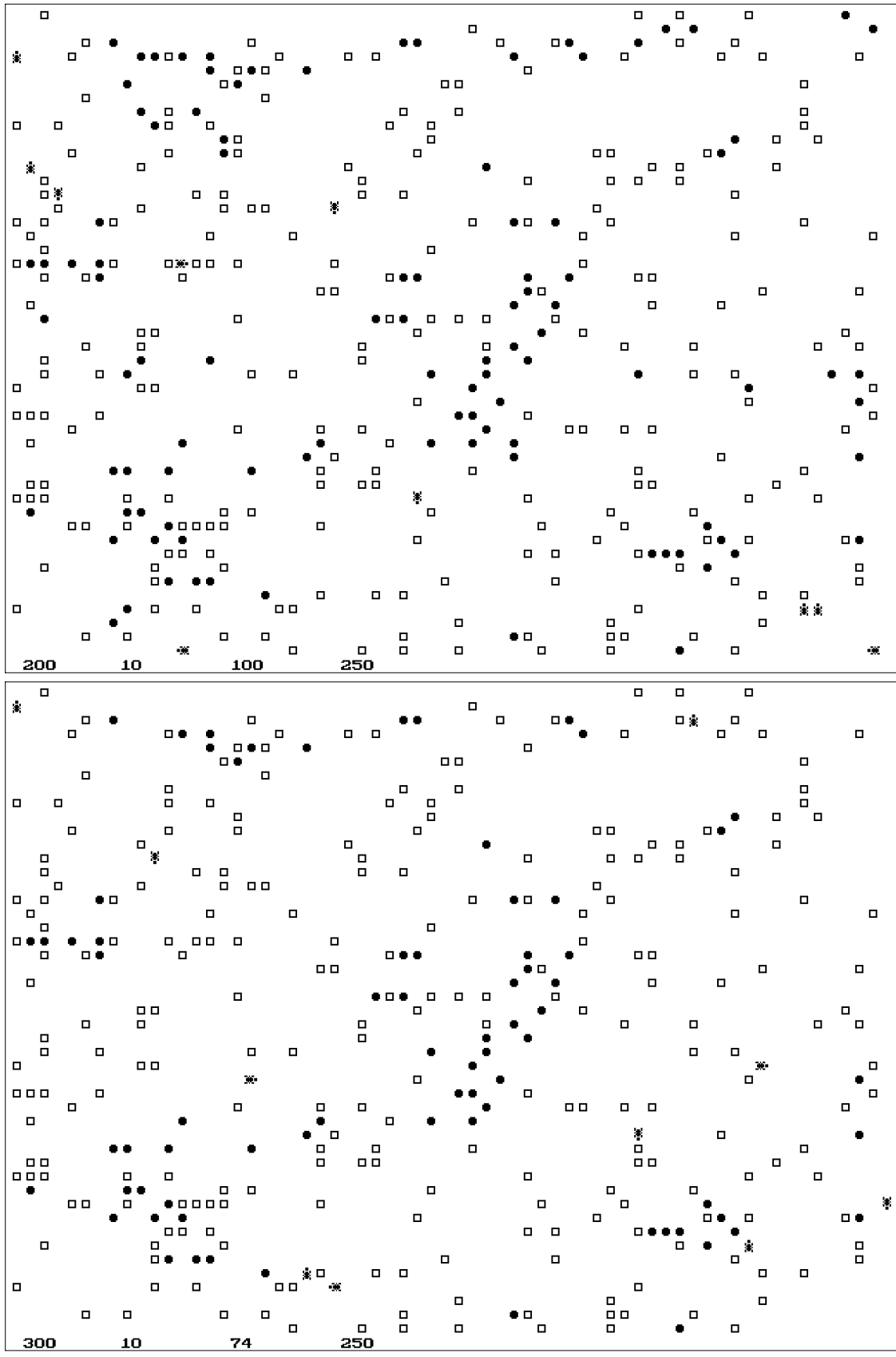
Obrázky 9e, 9f



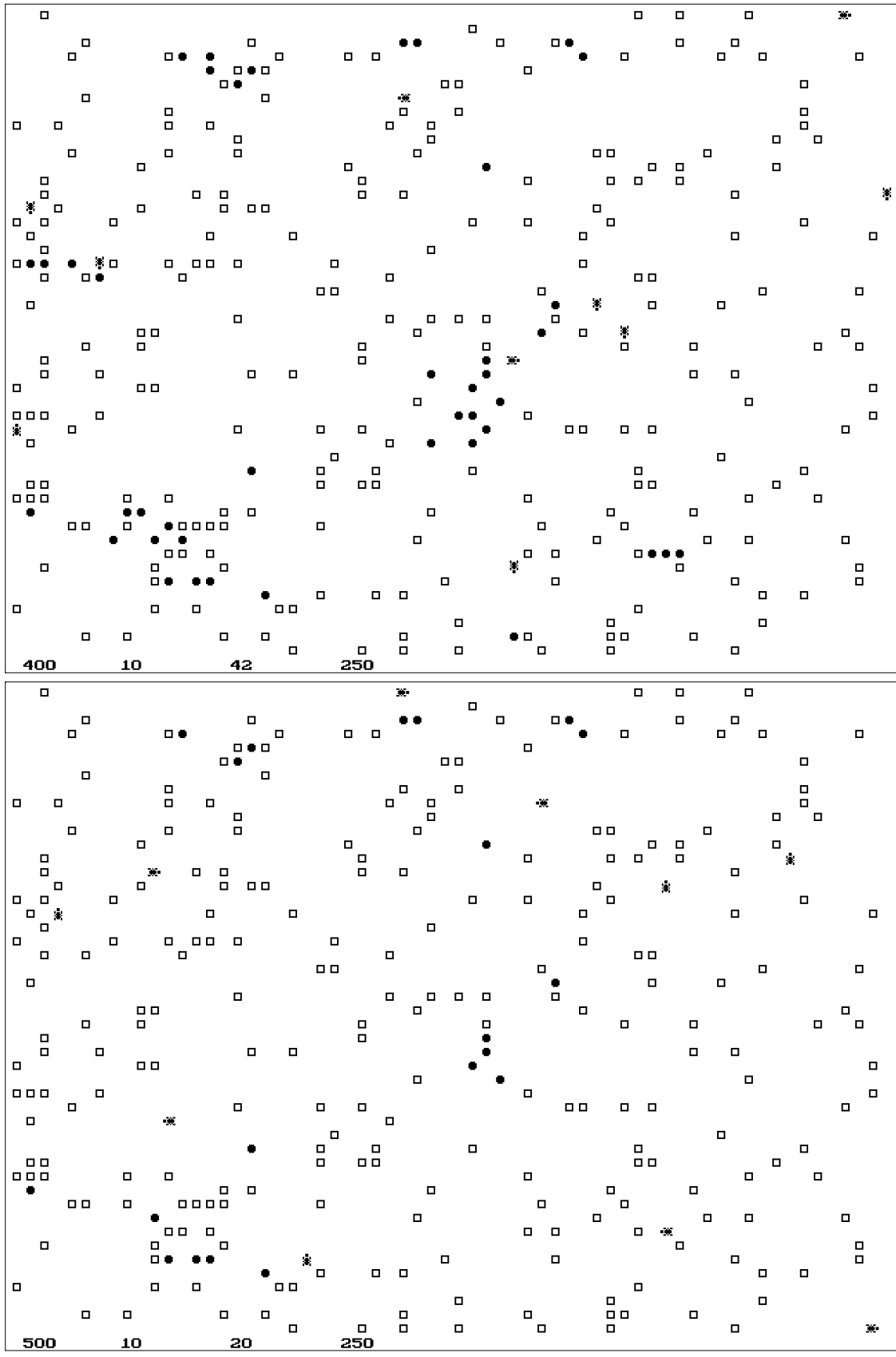
Obrázky 9g, 9h



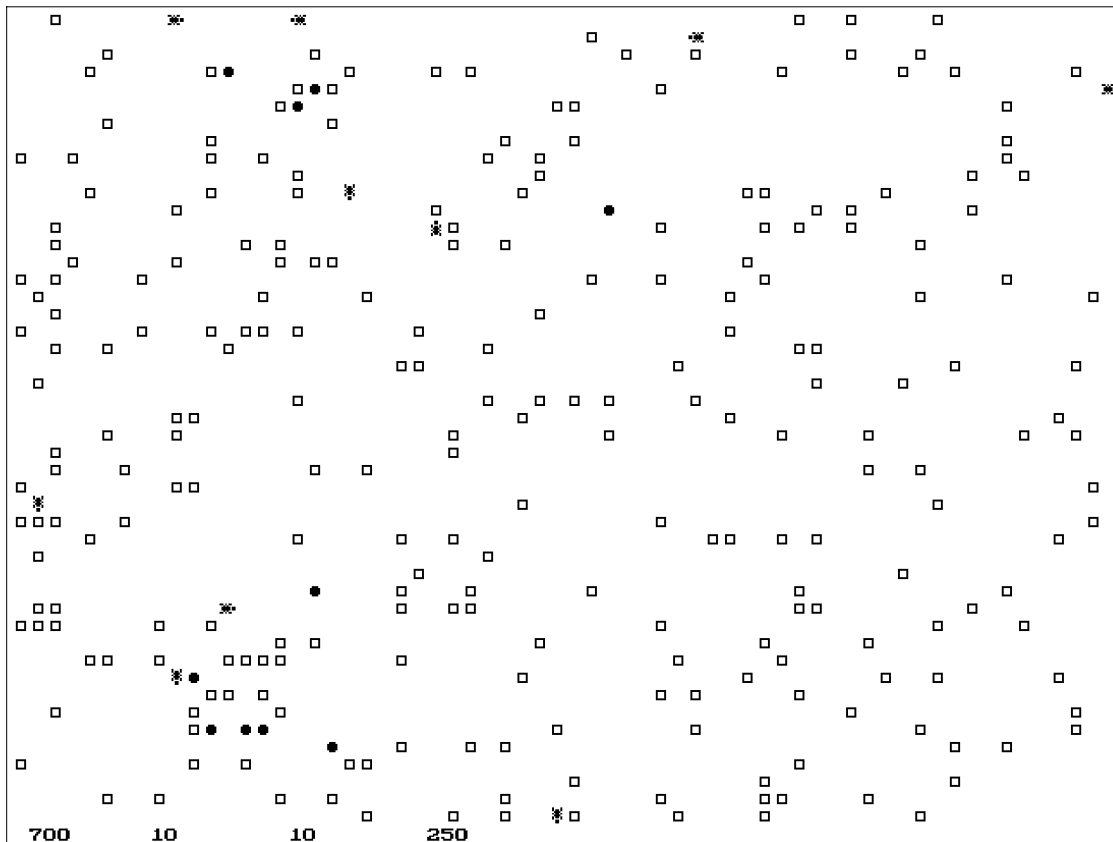
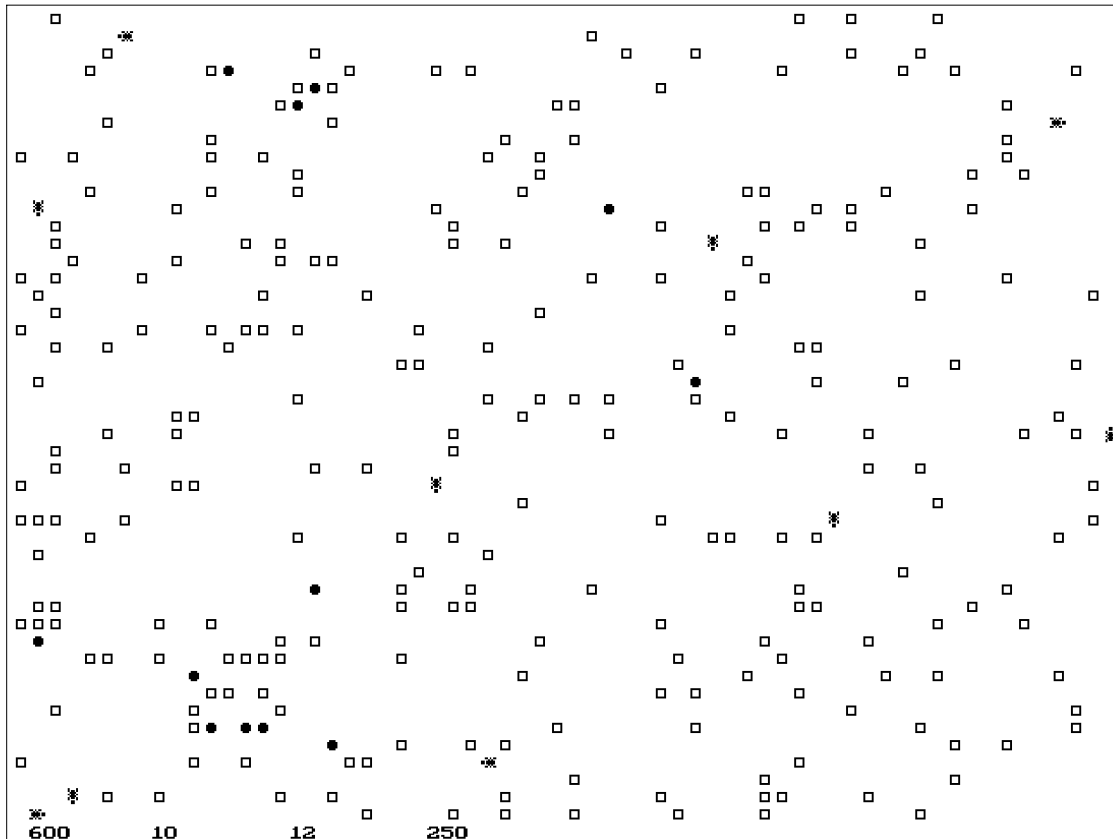
Obrázky 10a, 10b



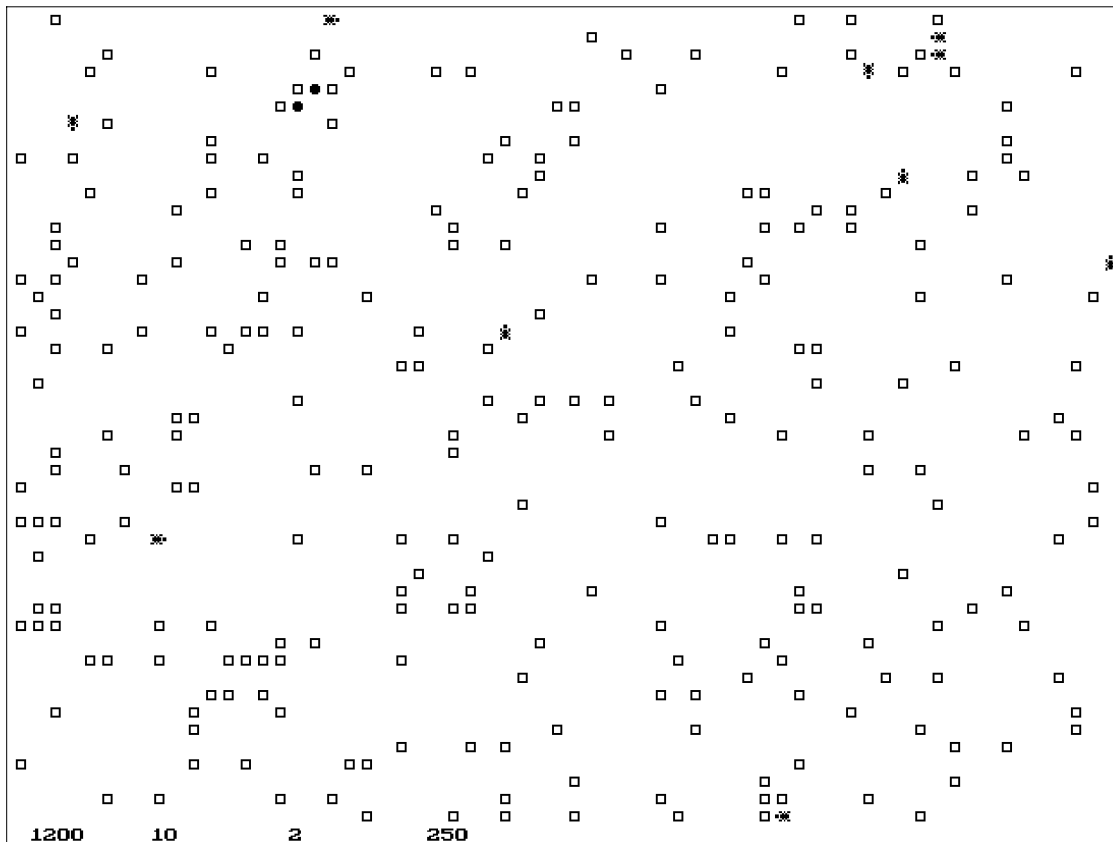
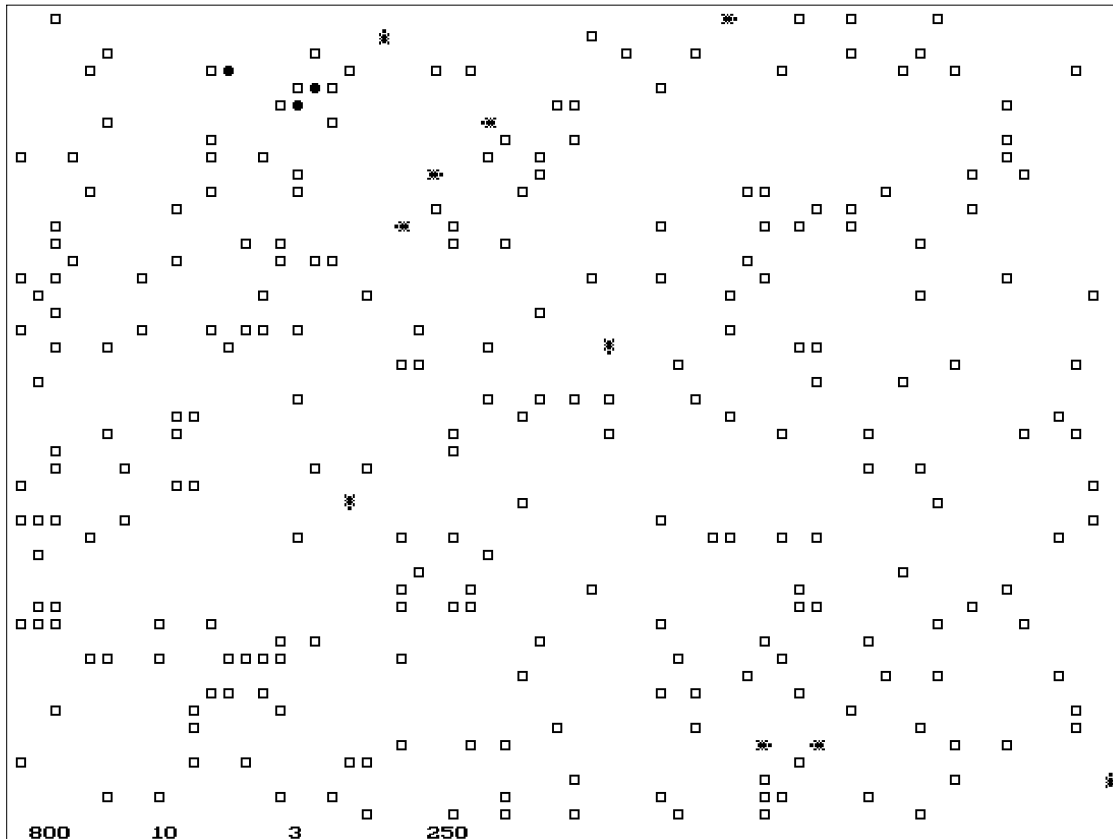
Obrázky 10c, 10d



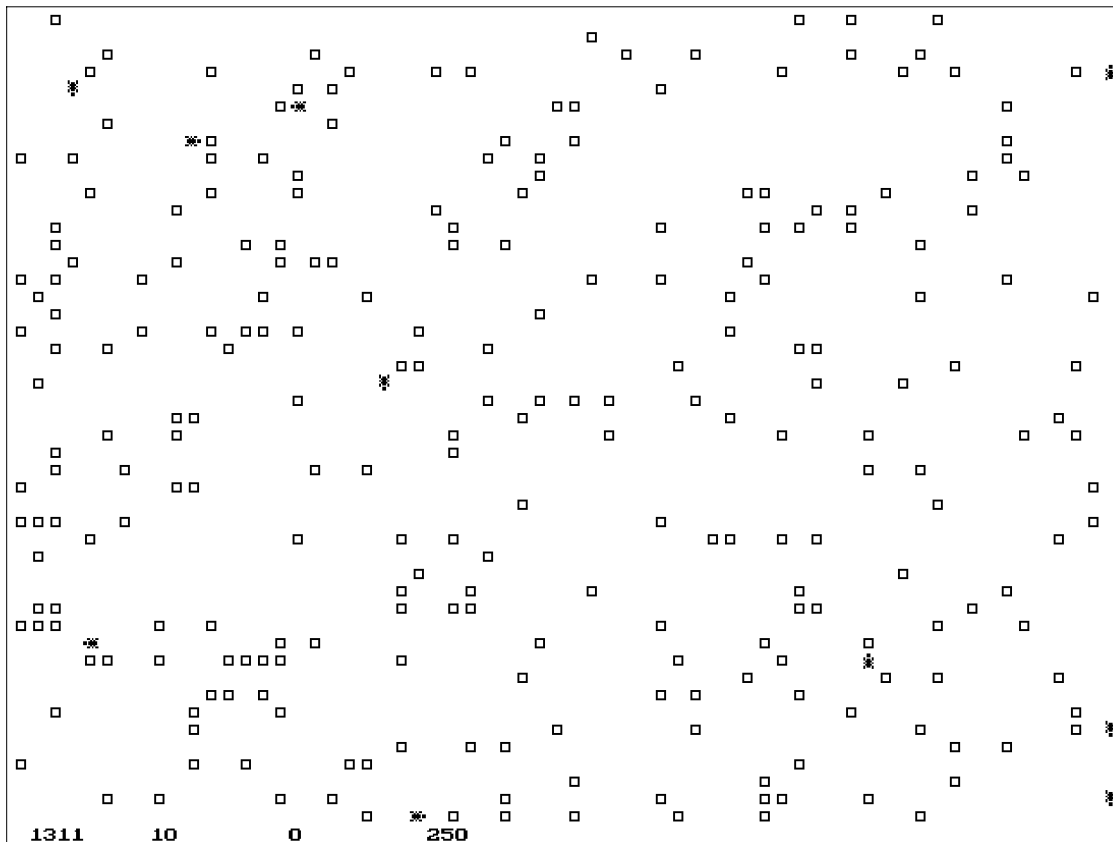
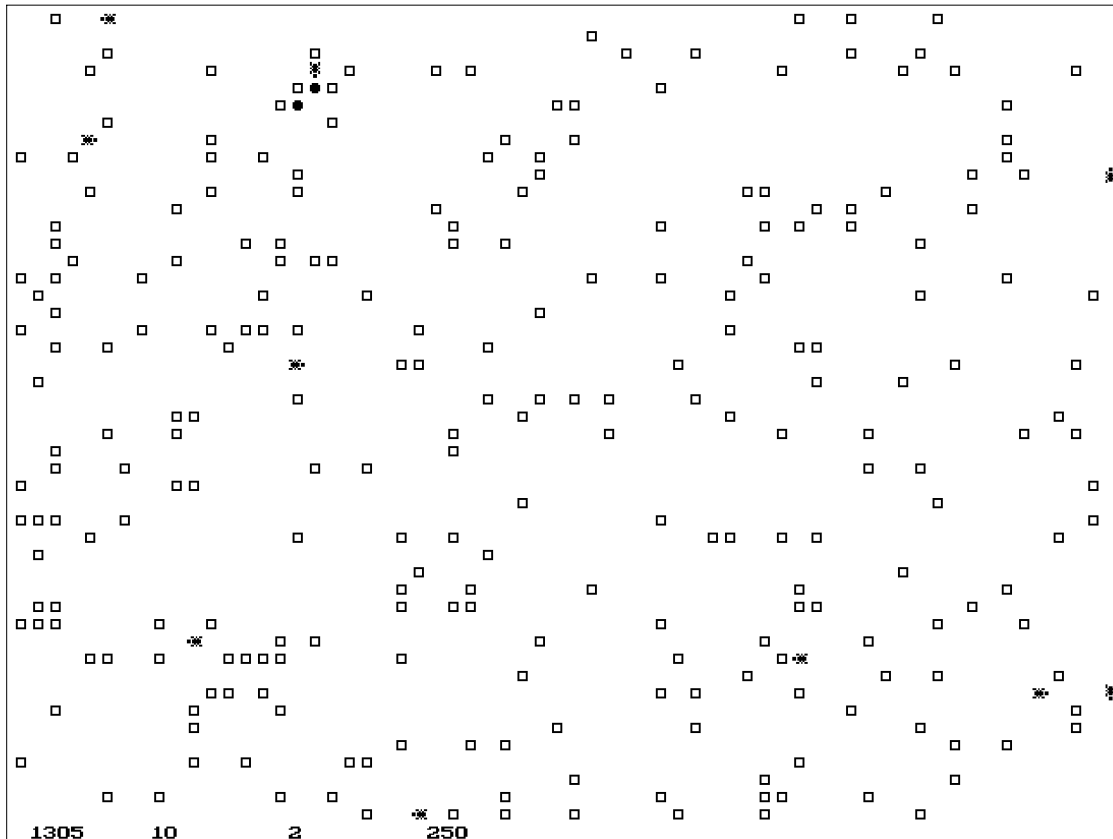
Obrázky 10e, 10f



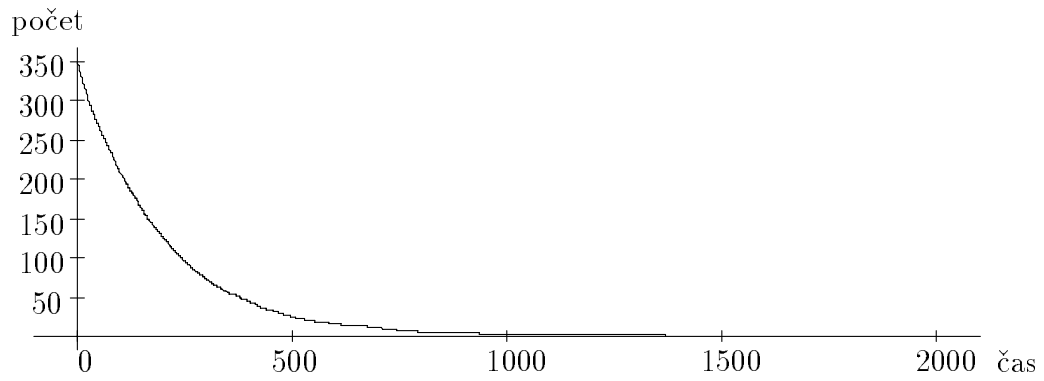
Obrázky 10g, 10h



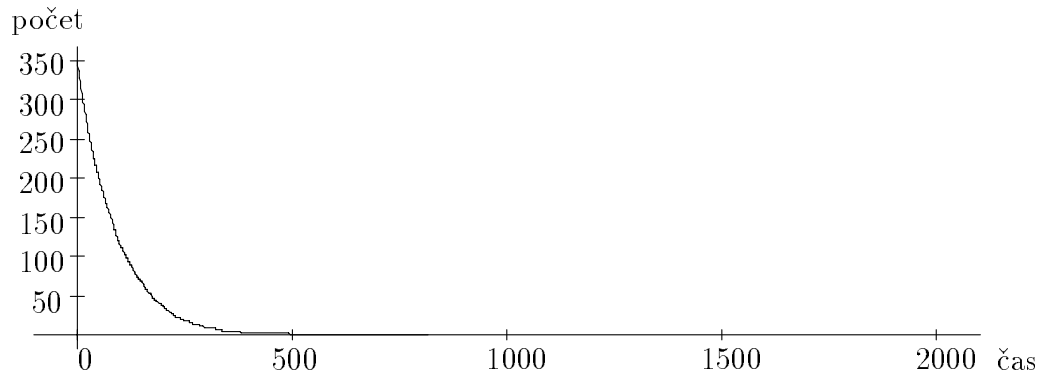
Obrázky 10i, 10j



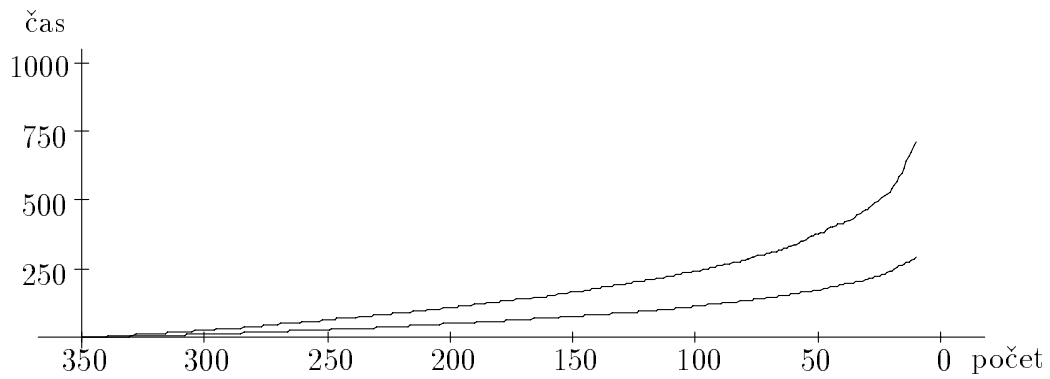
Obrázky 10k, 10l



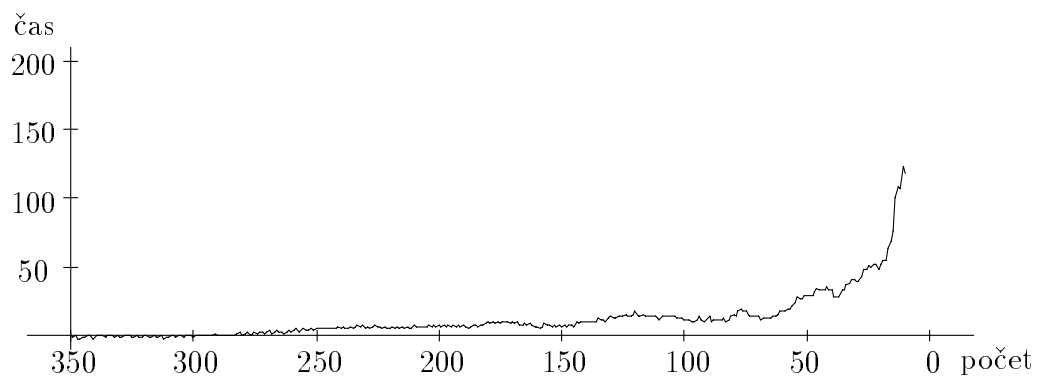
Obrázok 11



Obrázok 12



Obrázok 13



Obrázok 14

Meraniami aproximované funkcie f_1 a f_2 vidíme na obrázkoch 11 a 12. Ďalší obrázok 13 zobrazuje naraz funkcie f_1^{-1} , f_2^{-1} . Krivka kompetencie, vypočítaná podľa týchto funkcií, je na obrázku 14. Na nej vidíme, že v okolí medze 10 je hodnota cc výrazne kladná. Navyše hodnota cc má rastúcu tendenciu, čo vylučuje, že by išlo o náhodu alebo chybu. Na základe kritéria kompetencie môžeme teda prehlásiť, že kolónia dezaktivuje zamorené územia **kompetentne**. Schopnosť dezaktivácie sa u jednotlivých agentov zvyšuje s narastaním ich počtu. Zvyšovaním počtu agentov teda vzniká **emergentný efekt: narastanie ich kolektívnej zručnosti**.

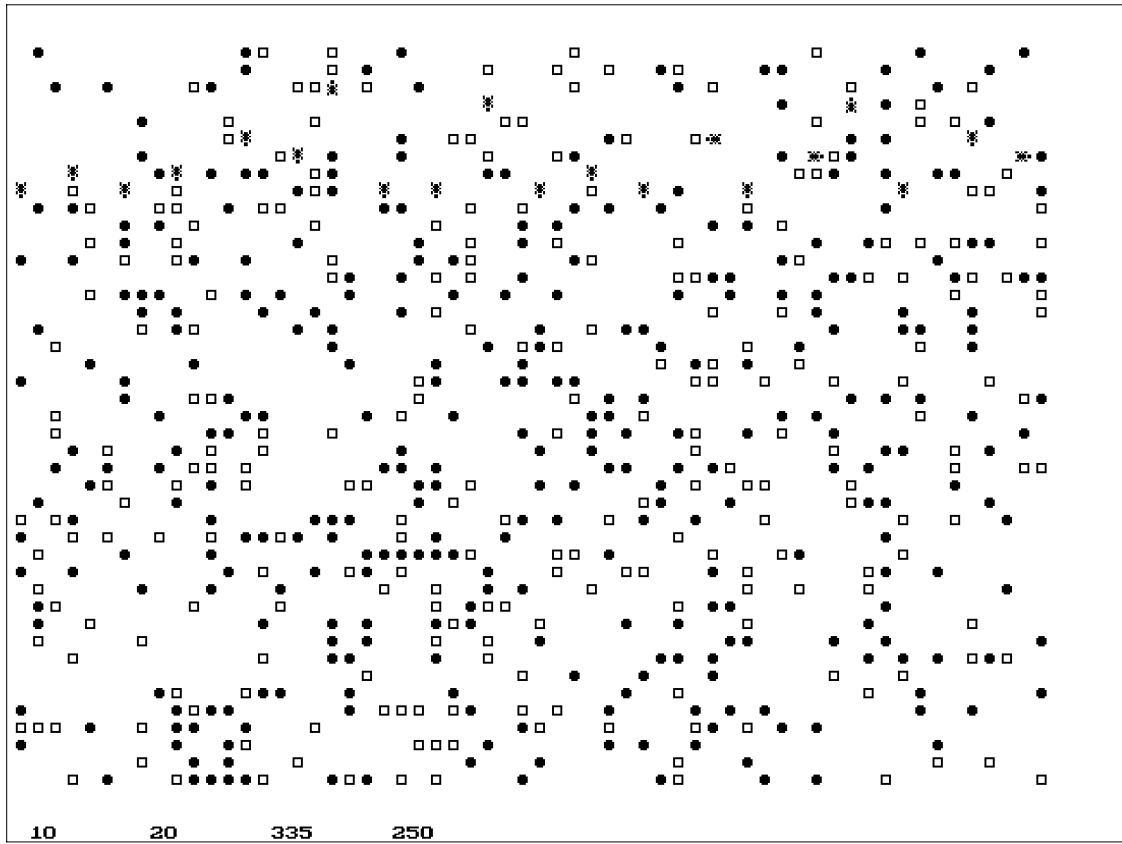
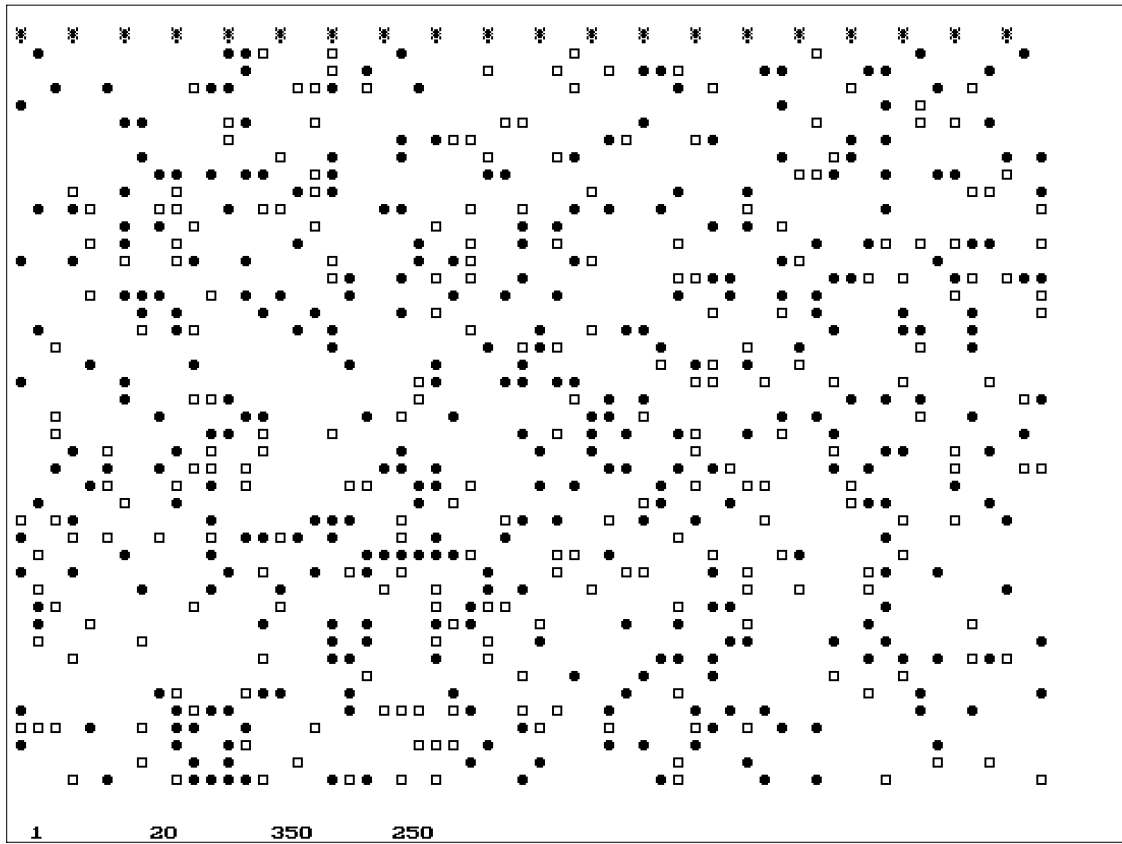
Čím je spôsobený tento efekt? Pravdepodobne tým, že pri lineárnom náraste počtu agentov nelineárne narastá počet ich zrážok (a teda i intenzita ich interakcie), čo má za následok nelineárny nárast veľkosti triedy trajektórií po ktorých sa agent pohybuje a teda aj intenzity dezaktivácie.

7.6 Kompetentní verzus nekompetentní

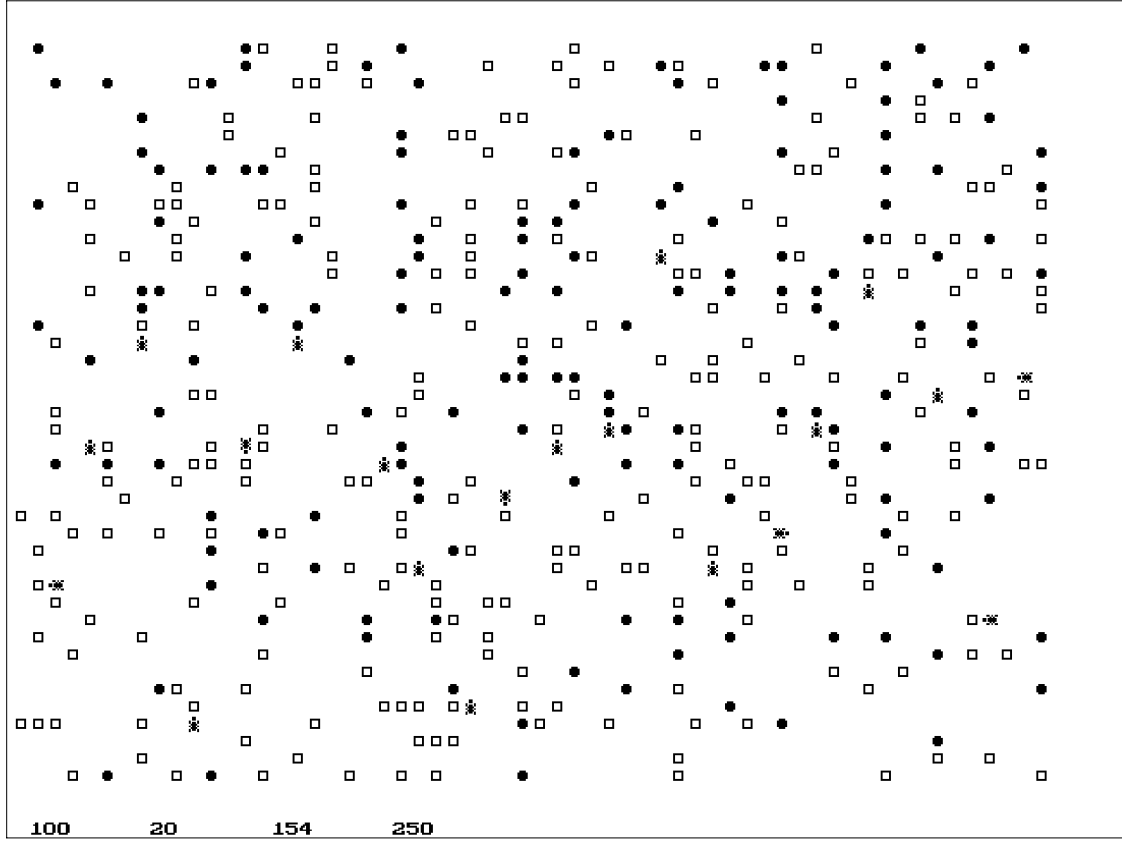
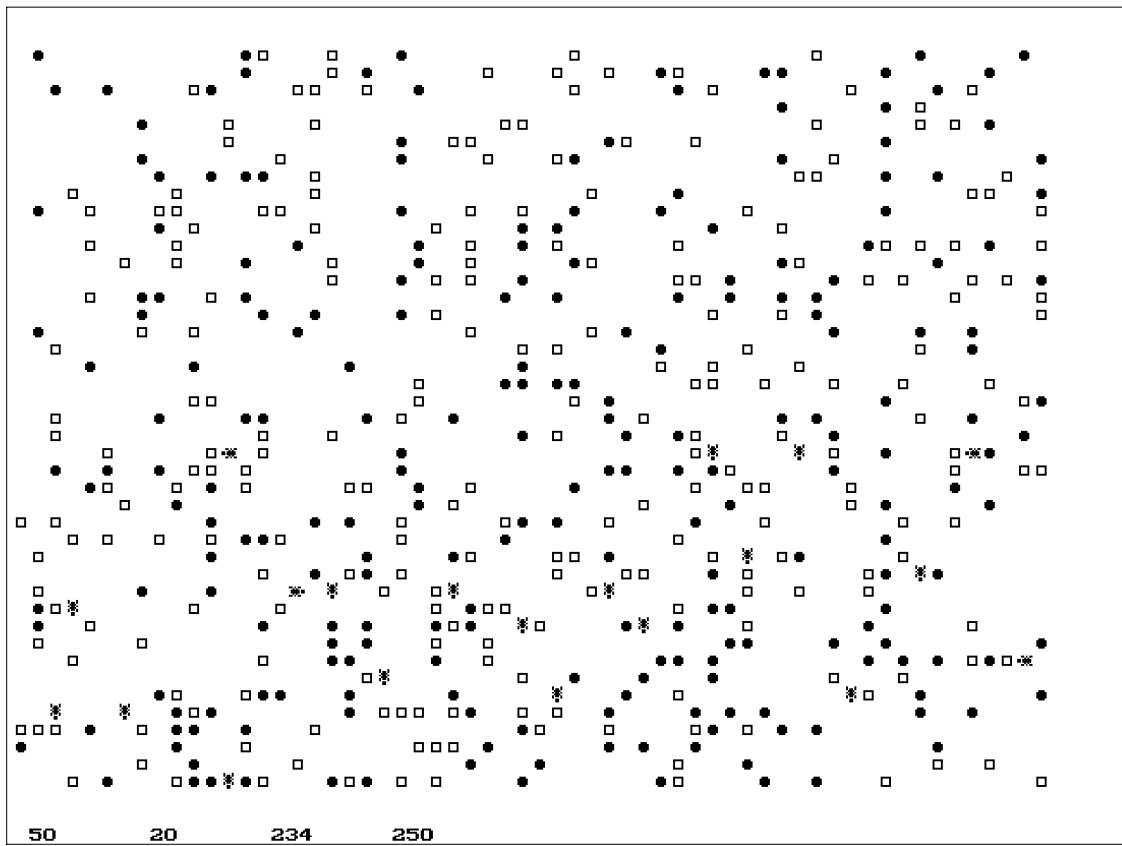
Uvažujúc o výsledkoch tohto experimentu sa vynára otázka, pre aký druh agentov by sme dostali iné výsledky. Akí agenti pracujú nekompetentne? Na zodpovedanie tejto otázky sme uskutočnili modelovania aktivít kolónie zloženej z iného druhu agentov. Pri tomto experimente agenti riešia v tom istom prostredí tú istú úlohu s tými istými schopnosťami (budú mať tie isté receptory a efekty) ale zásadne odlišným spôsobom.

Títo agenti budú mať tiež subsumpčnú architektúru, nie však jej špeciálnejšiu verziu používanú v tejto práci: nebudú to teda stochastickí agenti⁹. Nebudú ani čisto reaktívni, naopak, budú veľmi využívať vnútornú pamäť. S pomocou tejto pamäte budú poznať svoje absolútne súradnice a absolútny smer v priestore. V počiatočnej momentke budú pravidelne umiestnení na špeciálnych miestach, čo budú potrebovať k riešeniu úlohy. Konkrétne to budú miesta s najmenšou x -ovou súradnicou, pričom rozostupy medzi agentami budú rovnako veľké. Každý jeden agent bude schopný plánovať svoj pohyb, ktorý spočíva v prechádzaní priestoru po miestach s rovnakou x -ovou súradnicou. Pokiaľ narazí na prekážku, snaží sa ju obísť a vrátiť sa na pôvodnú x -ovú súradnicu. Chodí teda ako traktor keď orie, len s tým rozdielom, že mu v ceste môžu stáť prekážky, ktoré treba obísť. Tieto prekážky obchádza na základe pravidla ľavej alebo pravej ruky podľa toho, akým absolútnym smerom sa pohybuje. Pravidlo obchádzania prekážky volí agent tak, aby prekážku obchádzal po miestach, cez ktoré ešte nešiel. Keď prejde väčšinu miest na zvolenej x -ovej súradnici X , presunie sa na miesta s x -ovou súradnicou $X + 1$. Skutočnosť, že prešiel väčšinu miest na zvolenej x -ovej súradnici agent spoznáva na základe narazenia na objekt stena.

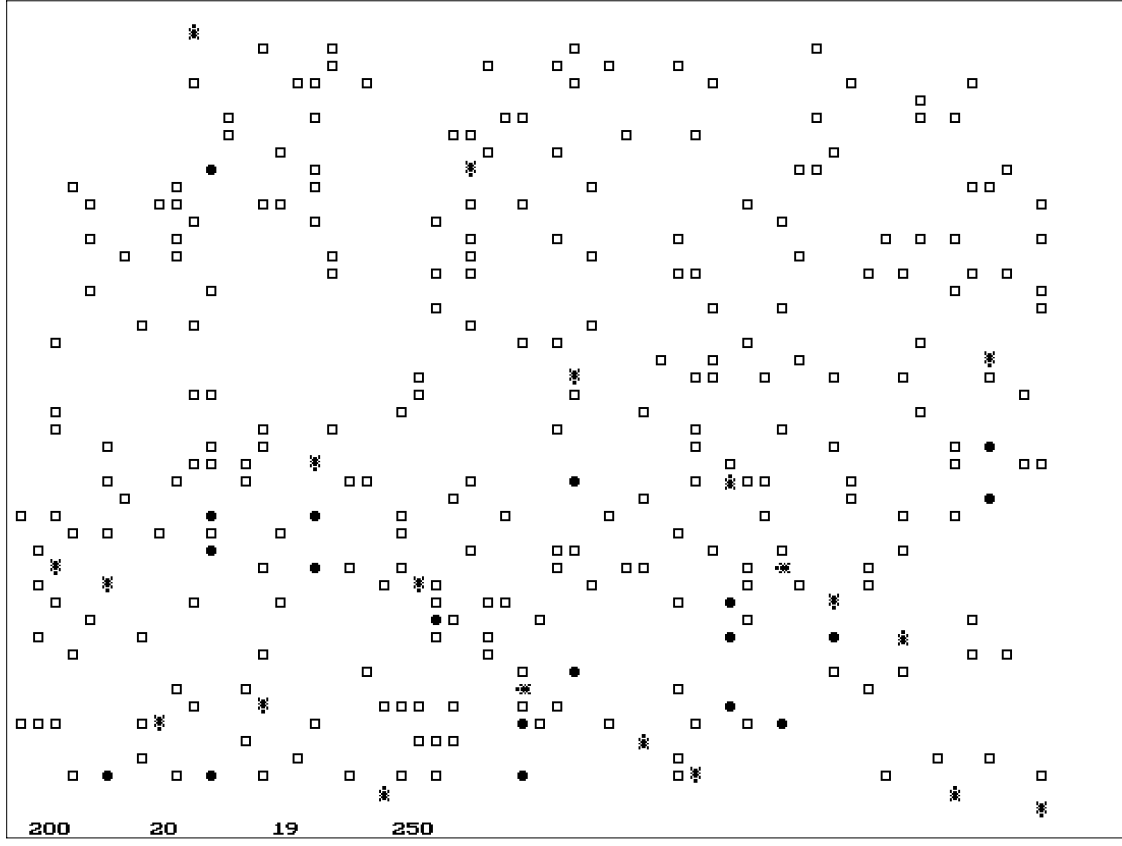
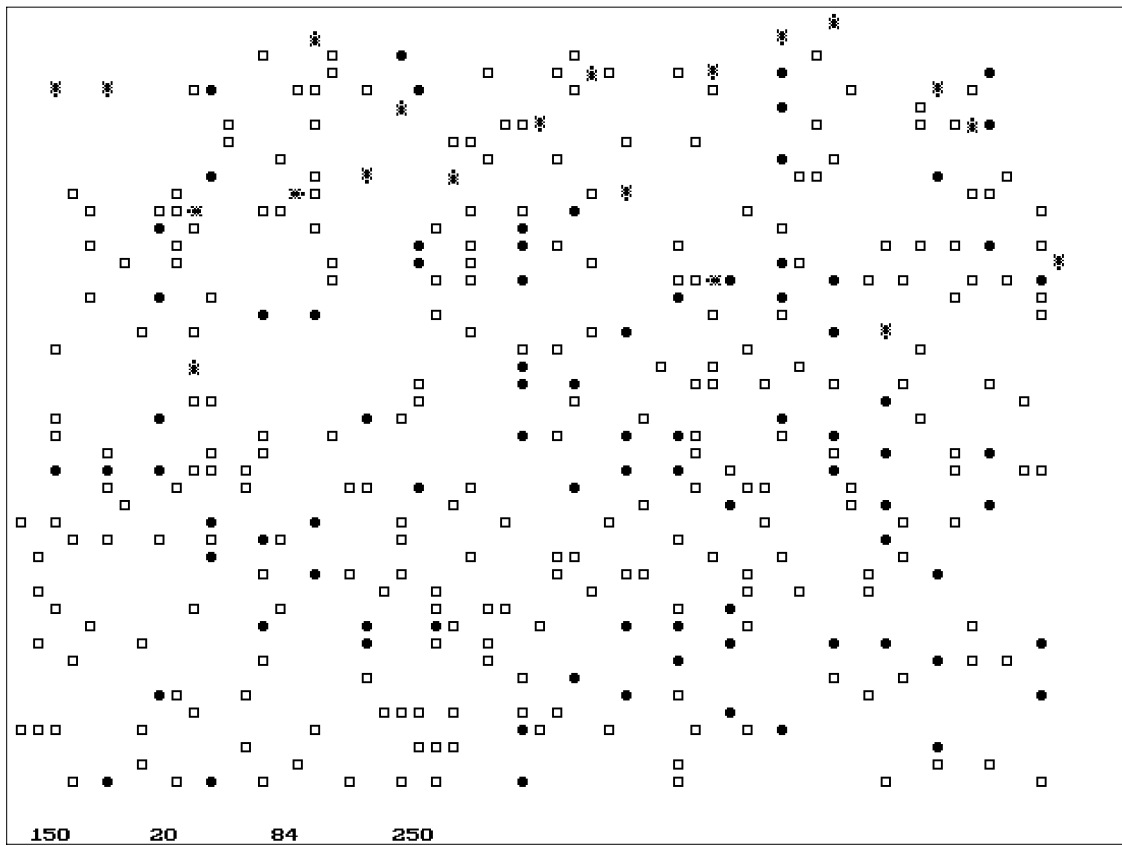
⁹ Prostredie **Mra** umožňuje modelovať agentov zodpovedajúcich akémukolvek procesu, teda aj takýchto agentov.



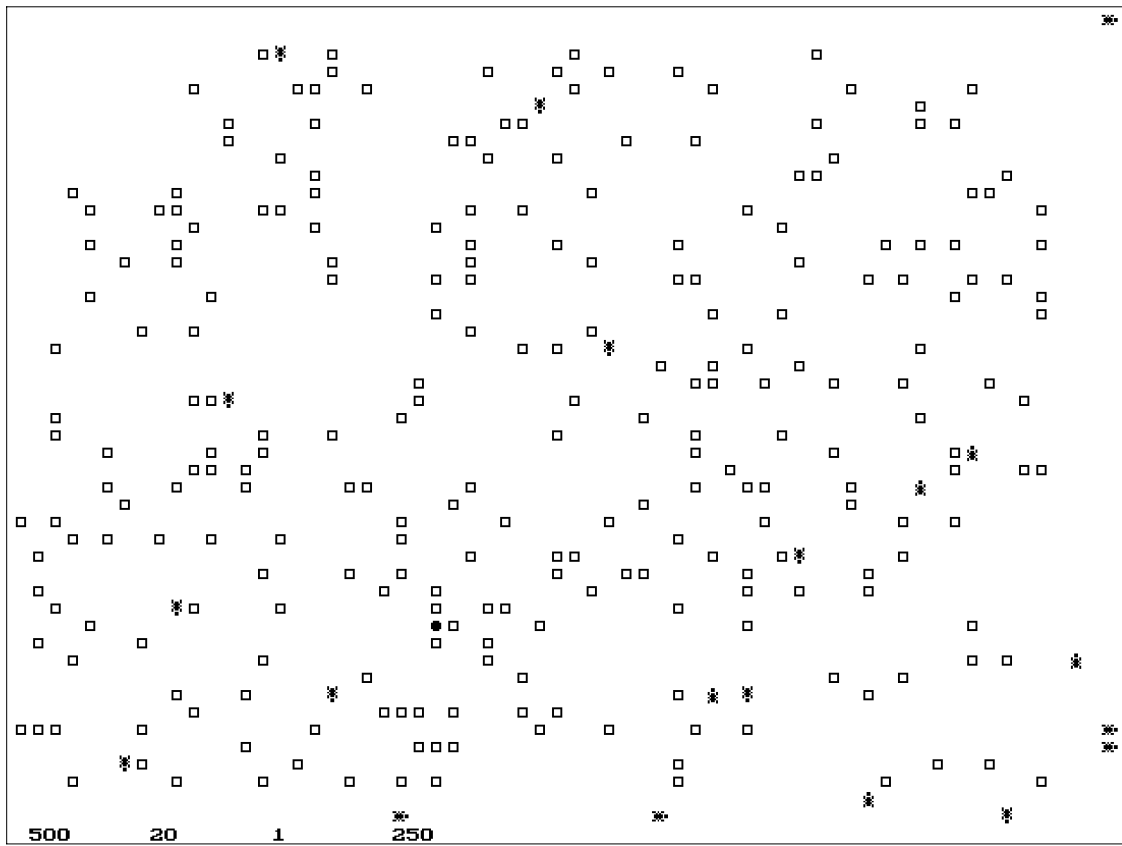
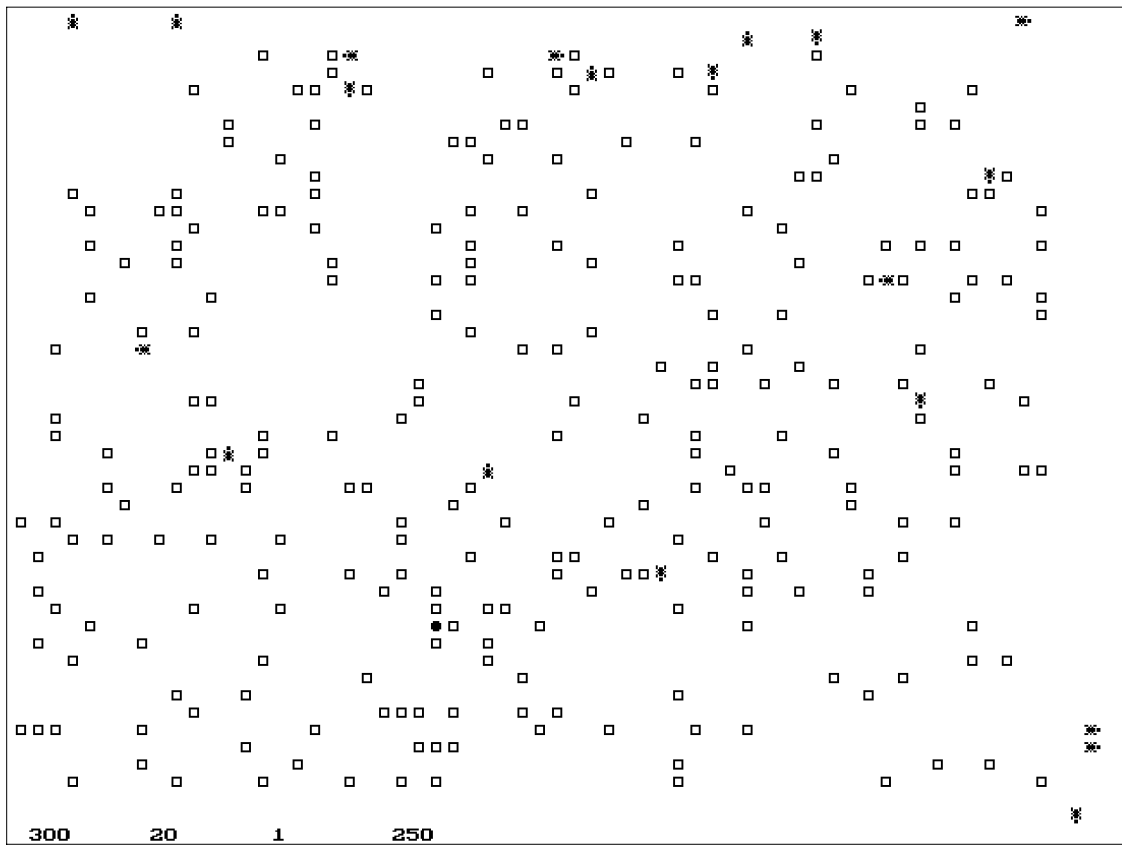
Obrázky 15a, 15b



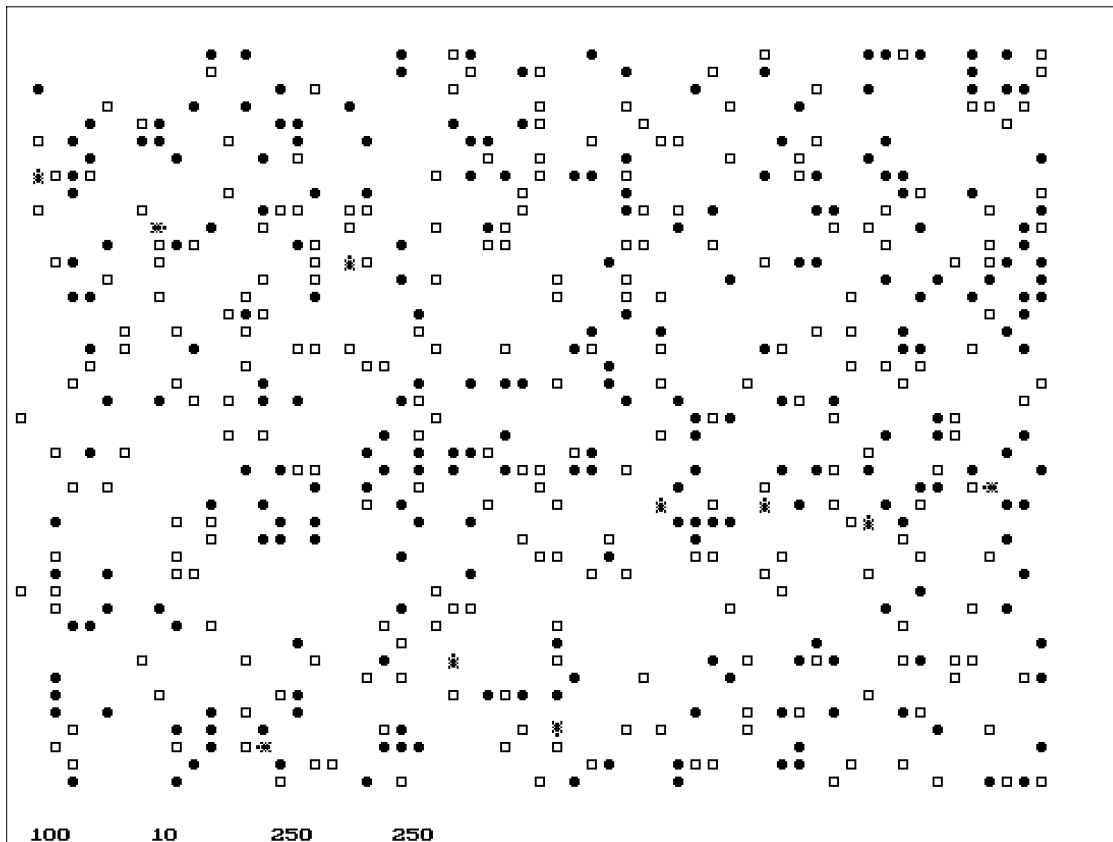
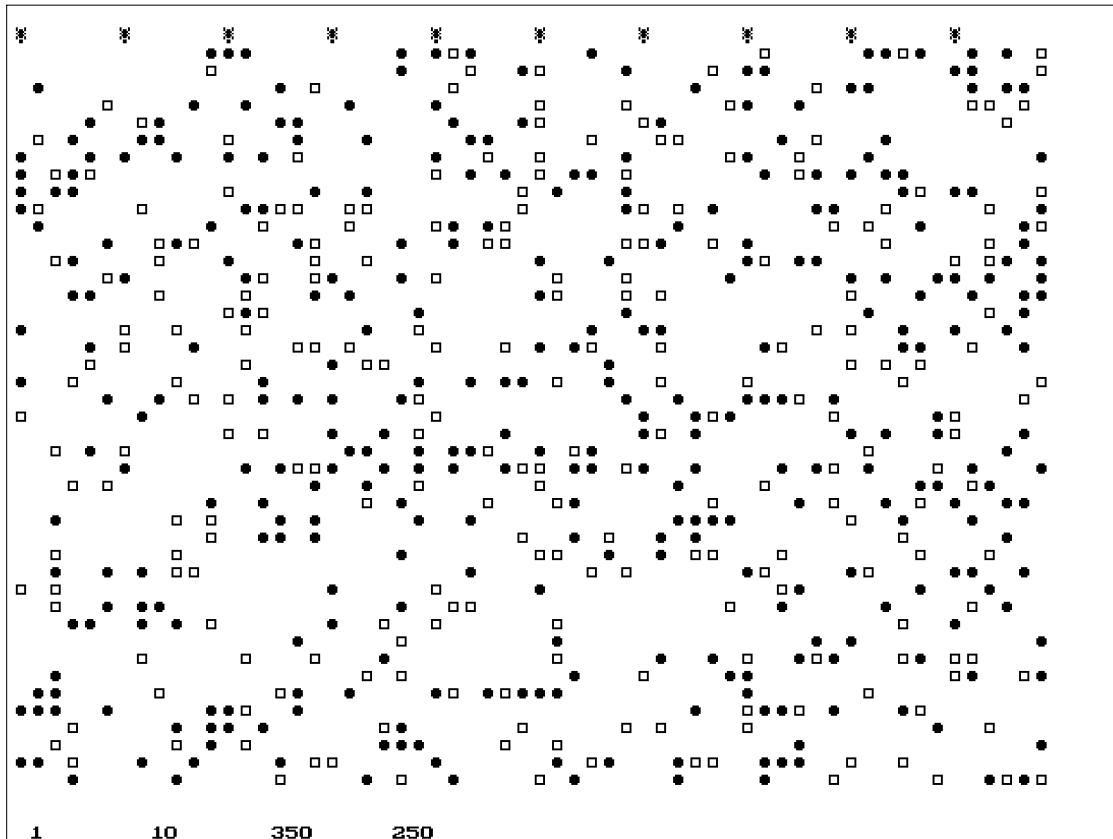
Obrázky 15c, 15d



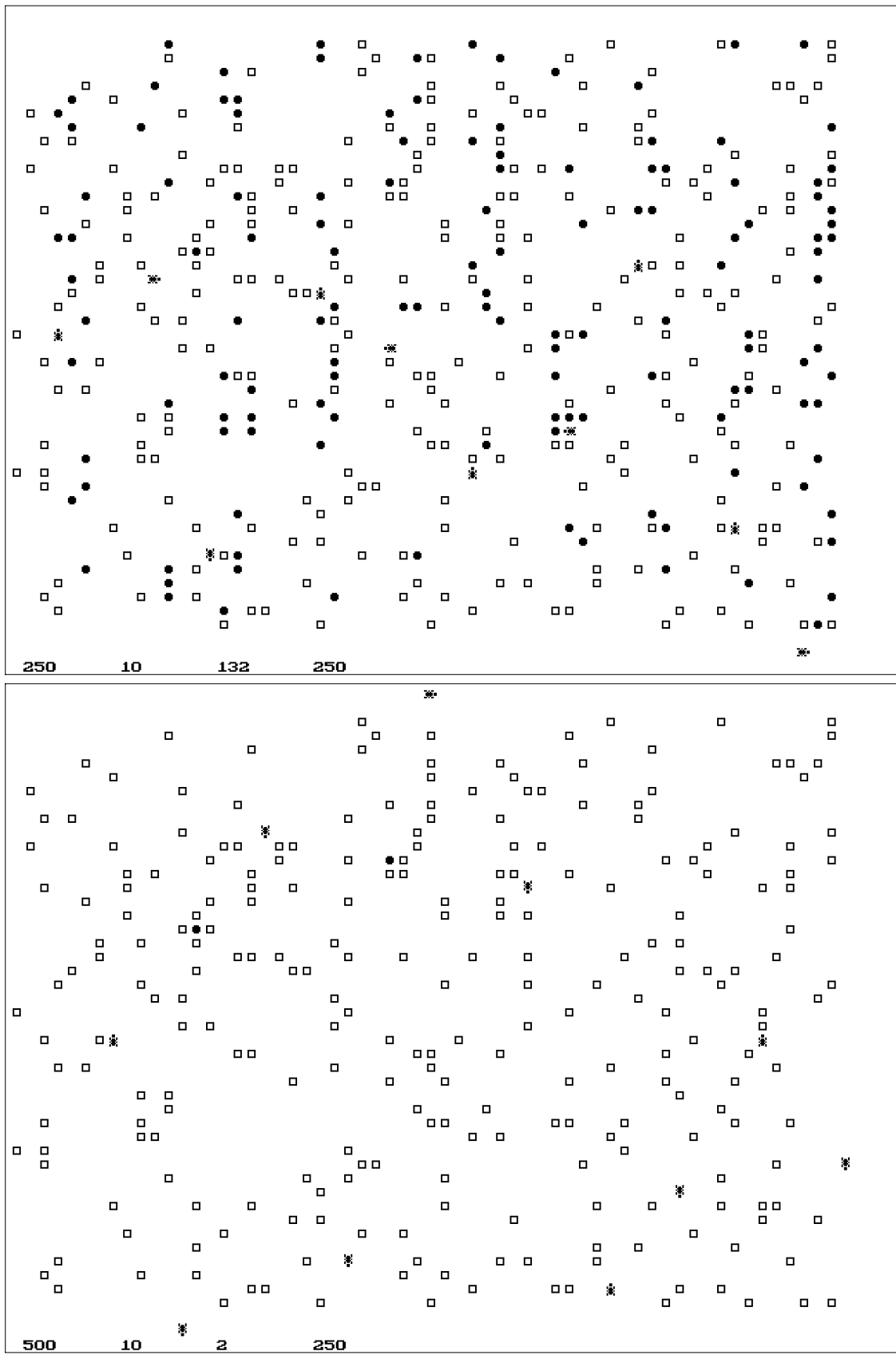
Obrázky 15e, 15f



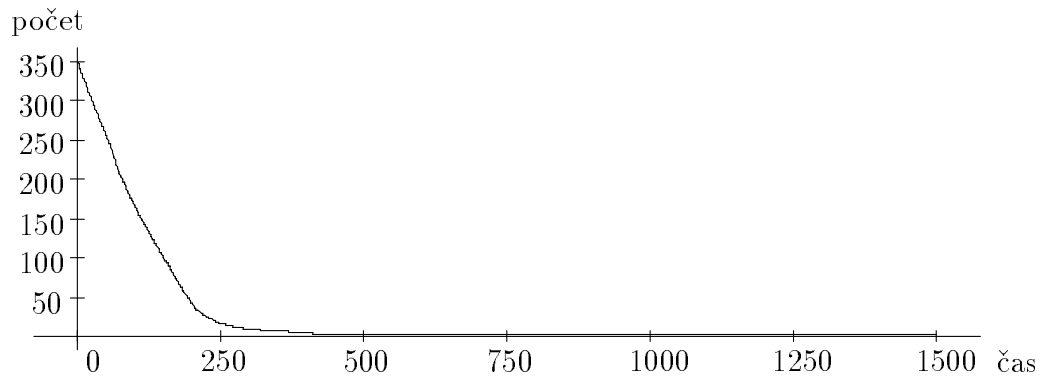
Obrázky 15g, 15h



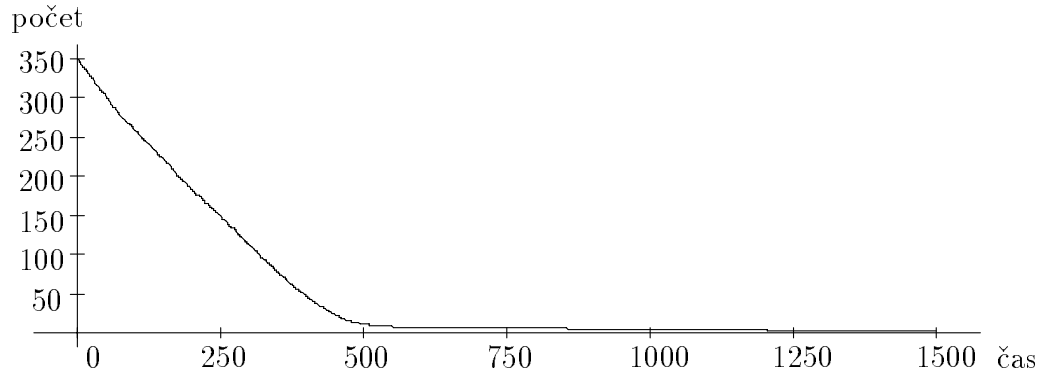
Obrázky 16a, 16b



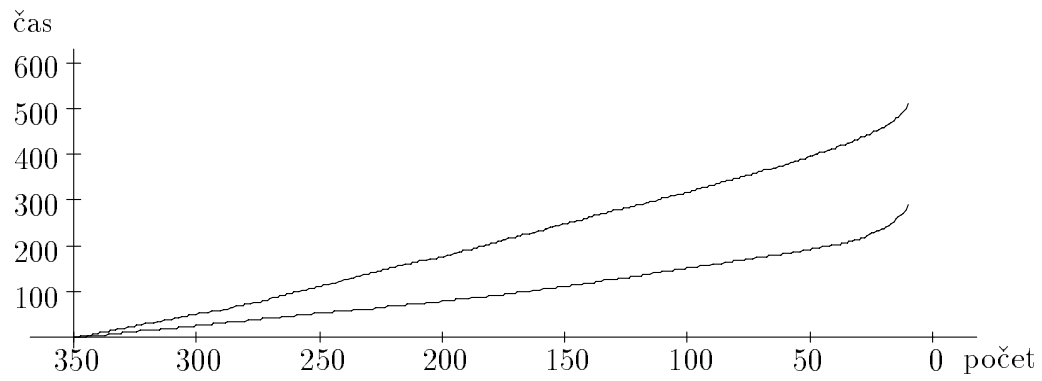
Obrázky 16c, 16d



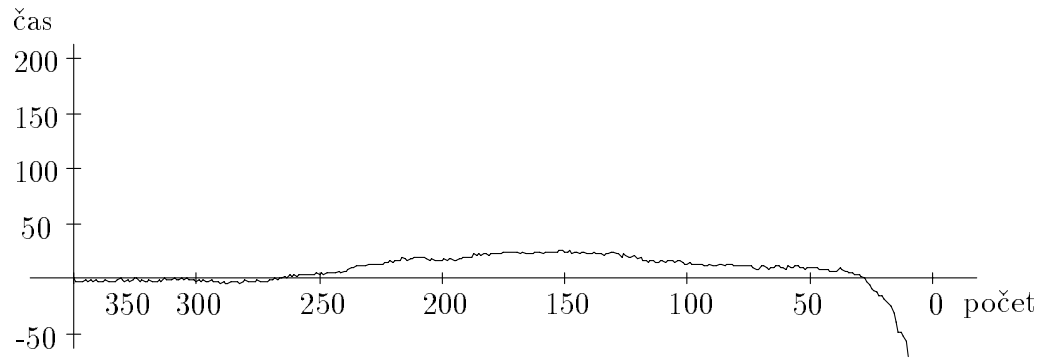
Obrázok 17



Obrázok 18



Obrázok 19



Obrázok 20

Bližší popis tohto agenta neuvádzame, lebo sme si na to v práci nevypracovali príslušný pojmový aparát a jeho vypracovanie prekračuje rozsah tejto práce.

Pozorovania ukázali, že na umiestnení objektov prostredia príliš nezáleží, ale nesmú byť pri stenách, aby neznemožnili agentom detekciu ukončenia pohybu na miestach s určitou x -ovou súradnicou. V záujme neskreslenia výsledkov je vhodné, aby na jedného agenta pripadalo rovnako veľa miest, takže na miesta zodpovedajúce zvyšku po delení počtu stĺpcov počtom agentov nebudeme umiestňovať zamorené územia. Počet agentov nesmie byť príliš veľký, aby si agenti „nezavadzali“. Počet prekážok nesmie byť príliš veľký, lebo potom by súvislé útvary z kameňov príliš bránili agentom vrátiť sa pri ich obchádzaní na pôvodnú x -ovú súradnicu. Na základe týchto skutočností sa ukázalo, že pre merané modelovanie možno vybrať kolóniu a prostredie s tými istými parametrami ako v prípade stochastických agentov. Prostredie bude teda obsahovať 250 kameňov a 350 zamorených území, kolónia sa bude skladať z 20-tich agentov a jej $1/2$ -kolónia z 10-tich agentov.

Pre kolóniu a $1/2$ -kolóniu sme uskutočnili spolu 10 meraných modelovaní. Ukážku vývoja kolónie môžeme sledovať na obrázkoch 15a-15h. Jeden z vývojev $1/2$ -kolónie môžeme sledovať na obrázkoch 16a-16d. Význam obrázkov 17, 18, 19, 20 je analogický významu obrázkov 11, 12, 13, 14.

Pri opise činnosti agentov sme si všimli, že agenti dezaktivujú zamorené miesta spôsobom, ktorý je veľmi blízky ľudskému uvažovaniu. Merania uskutočnených modelovaní však jednoznačne ukazujú, že takýto spôsob dezaktivácie zamorených území je **nekompetentný** (obrázok 20). Čím je spôsobený tento fakt? Pravdepodobne absenciou interakcie medzi agentami. Agenti sa síce počas obchádzania prekážok môžu stretať (a aj sa stretávajú), ale toto stretnutie nijako nemení ich činnosť, držia sa stále svojho plánu, ktorý spočíva v návrate na pôvodnú x -ovú súradnicu a stretnutý agent sa pre nich stáva iba ďalšou prekážkou, ktorú treba obísť, aby sa lokálny cieľ tohto plánu uskutočnil. Stretnutý agent nepredstavuje teda žiadny nový podnet, naopak, spôsobuje zdržanie vo vykonávaní pôvodného plánu obchádzania prekážky, čo sa na krivke kompetencie prejaví tým, že nie je v okolí medze nulová, ale záporná.

Pri kolóniách zložených z takýchto agentov teda nedochádza k emergentným efektom a použitie väčšieho počtu agentov spôsobuje len absolútne, ale nie relatívne zlepšenie ich činnosti.

7.7 Výsledky experimentu

Tento experiment ukázal, že spôsoby riešenia úloh bežne ľuďmi používané nemusia byť vhodné pre kolóniu agentov. Naopak, spôsob riešenia, ktorý je ľuďmi

považovaný za priveľmi založený na náhode, môže byť pre kolóniu tým pravým spôsobom, ktorý z nej urobí viac než len kompozíciu agentov. Experiment ďalej ukázal, že veľmi jednoduchá interakcia môže viesť k vzniku emergentných efektov. Ďalej bol pozorovaný fakt, že emergenté efekty sa v aktivitách kolónie prejavujú pri primerane zložitom prostredí. Nefunguje tu teda bežná úvaha, že čím je problém jednoduchší, tým ľahšie sa vyrieši a čím je zložitejší, tým to pôjde ťažšie. Zdá sa, že v príliš jednoduchých alebo v príliš zložitých prostrediach bude pre kolóniu riešenie úlohy ťažšie ako v primerane zložitých.

Kapitola 8

Experiment: Zbieranie smetí

8.1 Prostredie

Prostredie obsahuje dva druhy objektov: kamene a smeti. Na označenie identifikačných čísel ich druhov budeme používať konštanty `stone` a `rubbish`. Tieto objekty sú v počiatočnej momentke náhodne umiestnené v priestore. Rozmery prostredia sú 64×47 .

8.2 Schopnosti agentov

Agenti sú schopní vnímať svoje bezprostredné okolie a rozpoznávať v ňom kamene a smeti (odpadky). Ďalej sú schopní pohybovať sa smere svojej orientácie a otáčať sa. Majú chápadlo pomocou ktorého sú schopní uchopiť odpadok nachádzajúci sa v ich bezprostrednom okolí, niešť ho a položiť na voľné miesto. Agenti nedokážu premiestňovať alebo ničiť kamene, nedokážu niešť naraz viac odpadkov, ani uchopiť nejaký odpadok, keď už iný odpadok nesú.

8.3 Úloha

Úlohou kolónie agentov je smeti rozptýlené v priestore popremiestňovať tak, aby vznikli v priestore oblasti kde bude veľa smetí (smetiská) a oblasti kde nebudú žiadne smeti (čistinky).

Vyhodnocovacia funkcia tejto úlohy bude zložitejšia ako v predchádzajúcom experimente. Predstavuje bodovanie prostredia, kde prostredie dostáva za každú oblasť

rozmerov 2×2 , 4×4 , 8×8 , 16×16 a 32×32 , kde sa nachádza nejaký odpadok, jeden bod. Vzhľadom na rozmery prostredia 64×47 , je najväčšia hodnota vyhodnocovacej funkcie 9958, najmenšia hodnota závisí od počtu smetí. Vyhodnocovacia funkcia je teda definovaná nasledovne:

$$e(s) = \sum_{v=1}^5 \sum_{\substack{1 \leq r \leq 65-2^v \\ 1 \leq c \leq 48-2^v}} \left[\left(\sum_{\substack{r \leq i < r+2^v \\ c \leq j < c+2^v \\ \text{miesto}(a)=(i,j) \\ a \in s}} [\text{druh}(a) = \text{rubbish}] \right) > 0 \right]$$

kde $[\varphi] = \begin{cases} 1 & \text{ak } \varphi \text{ je splnená} \\ 0 & \text{ak } \varphi \text{ nie je splnená} \end{cases}$ (Evansova konvencia).

e nemusí byť a väčšinou nebýva na momentkách vývoja monotónna.

Medza tejto úlohy sa musí stanoviť na základe počtu smetí v prostredí. Vzhľadom na neskôr zvolené konkrétne prostredie zvolíme medzu $\nu = 5700$. Keby sme chceli mať jednotnú medzu pre všetky prostredia, museli by sme za vyhodnocovaciú funkciu zvoliť $\frac{e(s)}{\psi(p)}$, kde p je počet smetí v momentke s . Vzhľadom na to, že voľba funkcie ψ je otázna a problematická, používame radšej pohyblivú medzu úlohy.

8.4 Riešenie: zloženie agenta a kolónie

Kolónia navrhnutá na riešenie bude obsahovať agentov jediného druhu. Agent bude pozostávať:

- z receptorov `Straight`, `Left`, `Right`, `Back`, `HandStatus`, `Rnd`, `Rnd2`
- z efektorov `Move`, `Turn`, `Hand` a
- z behaviorálnych modulov `Moving2`, `Avoidance2`, `FindingRubbish`, `TakingRubbish`, `FindingDump`, `PuttingRubbish` a `CrossingDump`.

Receptor `HandStatus` indikuje stav na efektore `Hand`. Vysiela dva druhy signálov. Signál `free` vysiela vtedy, keď sa v chápadle nenachádza žiadny odpadok. V opačnom prípade vysiela signál `occupied`. `HandStatus` :

$$r(s, m, d) = \text{stav-efektorov}(a)(\text{Hand}) = \begin{cases} \text{free} \\ \text{occupied} \end{cases}$$

kde $a \in s$ a $\text{miesto}(a) = m$.

Receptor `Rnd2` je analogický receptoru `Rnd`, líši sa iba v tom, že slúži ako zdroj podnetov pre iné aktivity.

Efektor `Hand` slúži na uchopenie, nesenie a polozenie odpadku. Jeho koproduktor je schopný prijať od behaviorálnych modulov signály `no`, `takestraight`, `putstraight`, `putleft`, `putback` a `putright`, ktoré spracúva do jedného na základe binárnej operácie ϕ_{Hand} , ktorá je definovaná v tabulke 4. Ak bude tento spracovaný signál `no`, efektor nevykoná žiadnu akciu, ak `takestraight`, efektor vykoná akciu, ktorej výsledkom bude uchopenie odpadku, ktorý sa nachádza v bezprostrednom okolí agenta v smere jeho orientácie a ak to bude jeden zo signálov `putstraight`, `putleft`, `putback` alebo `putright`, efektor vykoná akciu, ktorej výsledkom bude polozenie predmetu na jedno z miest v bezprostrednom okolí agenta. Tieto akcie budú samozrejme vykonané iba vtedy, keď budú vykonateľné.

ϕ_{Hand}	<code>no</code>	<code>takestraight</code>	<code>putstraight</code>	<code>putleft</code>	<code>putback</code>	<code>putright</code>
<code>no</code>	<code>no</code>	<code>takestraight</code>	<code>putstraight</code>	<code>putleft</code>	<code>putback</code>	<code>putright</code>
<code>takestraight</code>	<code>takestraight</code>	<code>takestraight</code>	<code>no</code>	<code>no</code>	<code>no</code>	<code>no</code>
<code>putstraight</code>	<code>putstraight</code>	<code>no</code>	<code>putstraight</code>	<code>no</code>	<code>no</code>	<code>no</code>
<code>putleft</code>	<code>putleft</code>	<code>no</code>	<code>no</code>	<code>putleft</code>	<code>no</code>	<code>no</code>
<code>putback</code>	<code>putback</code>	<code>no</code>	<code>no</code>	<code>no</code>	<code>putback</code>	<code>no</code>
<code>putright</code>	<code>putright</code>	<code>no</code>	<code>no</code>	<code>no</code>	<code>no</code>	<code>putright</code>

Tabuľka 4

Modul `Moving2` je podobný modulu `Moving` z kapitoly 6. Líši sa len tým, že s malou pravdepodobnosťou zmení smer agenta, pričom náhodne volí, či to bude otočenie doľava alebo doprava. Tieto smery však nie sú rovnocenné. V pomere 2:1 je zvýhodnený smer doľava, čo dáva agentovi ľavotočivú tendenciu.

Modul `Avoidance2` je podobný modulu `Avoidance` z kapitoly 6. Líši sa tým, že posilňuje ľavotočivosť agenta.

Podobne ako ľudia majú tendenciu byť praváci, títo agenti budú mať tendenciu ľavotočivosti. Ľavá strana im bude milšia ako pravá. Táto skutočnosť sa zrejme odrazí v triede trajektórií, po ktorých sa bude agent pohybovať a to asi tak, že sa zvýrazia prieniky tried trajektórií všetkých agentov. To agentom pomôže „dohodnúť sa medzi sebou“ kde bude smetisko a kde bude čistinka.

Modul `FindingRubbish` zabezpečuje vyhľadávanie smetí v bezprostrednom okolí agenta a výber, ktorá z týchto smetí bude uchopená a odnesená preč. Tento modul sa aktívne prejavuje len vtedy, keď sa v chápadle agenta nenachádza žiadny odpadok.

Modul `TakingRubbish` zabezpečuje uchopenie smeti, ktorá sa ocitne pred agentom. Tento modul je takisto aktívny len vtedy, keď sa v chápadle agenta žiadna smet nenachádza.

Modul `FindingDump` zabezpečuje vyhľadanie vhodného miesta na polozenie uchopeného odpadku. Aby si nevybral prvé ktoré nájde, volí niektoré z vhodných miest na základe náhody. To spôsobí, že agent sa bude z odpadkom trochu dlhšie prechádzať, kým ho niekam položí. Modul je aktívny len vtedy, keď sa v chápadle nachádza nejaký odpadok.

Modul `PuttingRubbish` zabezpečuje samotné polozenie uchopeného odpadku. Modul je takisto aktívny len vtedy, keď sa v chápadle nachádza nejaký odpadok.

Modul `CrossingDump` zabezpečuje správanie agenta, ktorý sa dostal do vnútra smetiska (ostatní agenti ho obhádzali smetami). Ak má agent v takej chvíli voľné chápadlo, môže sa dostať von aj sám (v opačnom prípade ho musí vyslobodiť iný agent, ak sa taký nenájde, má „zasypaný“ agent smolu). Práve tento modul spôsobuje prehrňanie sa smetami, až kým sa agent nedostane zo smetiska von.

Moving2

```

receptory: Rnd
efektory: Turn, Move
telo:      if Rnd < 1/40
              then
                if Rnd < 1/120
                  then Turn(right); Move(go)
                  else Turn(left); Move(go)
                else Turn(no); Move(go)

```

Avoidance2

```

receptory: Straight, Rnd
efektory: Turn
telo:      if Straight <> nothing
              then
                if Rnd < 1/3
                  then Turn(right)
                  else Turn(left)
                else Turn(no)

```

FindingRubbish

```

receptory: Straight, Left, Right, HandStatus
efektory: Turn, Move
telo:      if (HandStatus = free) and (Straight <> rubbish)
              then
                if Left = rubbish
                  then Turn(left!); Move(stop)
                  else
                    if Right = rubbish
                      then Turn(right!); Move(stop)
                      else Turn(no); Move(no)
                else Turn(no); Move(no)

```

TakingRubbish

```

receptory: Straight, HandStatus
efektory: Hand
telo:      if (HandStatus = free) and (Straight = rubbish)
              then Hand(takestraight)
              else Hand(no)

```

FindingDump

```

receptory: Straight, Left, Right, HandStatus, Rnd2
efektory: Turn, Move
telo:      if (HandStatus = occupied) and (Straight <> rubbish)
              then
                if Rnd2 < 1/3
                  then
                    if Left = rubbish
                      then Turn(left!); Move(stop)
                      else
                        if Right = rubbish
                          then Turn(right!); Move(stop)
                          else Turn(no); Move(go)
                    else Turn(no); Move(go)
                else Turn(no); Move(go)

```

PuttingRubbish

```

receptory: Straight, Left, Right, HandStatus, Rnd2
efektory: Hand, Turn, Move
telo:      if (HandStatus = occupied) and (Straight = rubbish)
              then
                if Rnd2 < 1/2
                  then
                    if Left = nothing
                      then
                        if Rnd2 < 1/4
                          then Hand(putleft); Turn(no!); Move(go!)
                          else Hand(putleft); Turn(back!); Move(go!)
                        else Hand(no); Turn(no!); Move(stop)
                      else
                        if Right = nothing
                          then
                            if Rnd2 < 3/4
                              then Hand(putright); Turn(no!); Move(go!)
                              else Hand(putright); Turn(back!); Move(go!)
                            else Hand(no); Turn(no!); Move(stop)
                        else Put(no); Turn(no); Move(go)

```

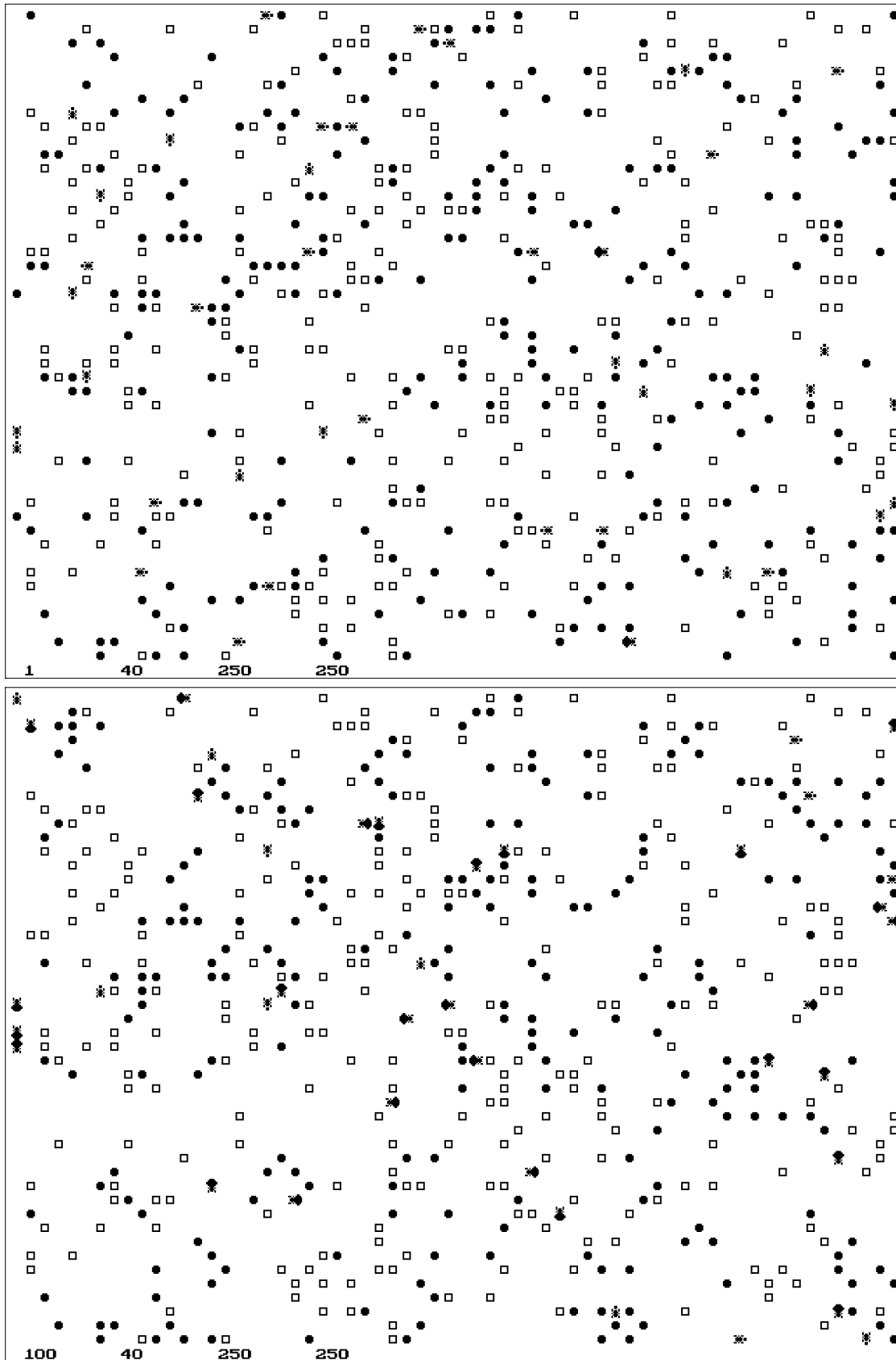
CrossingDump

```

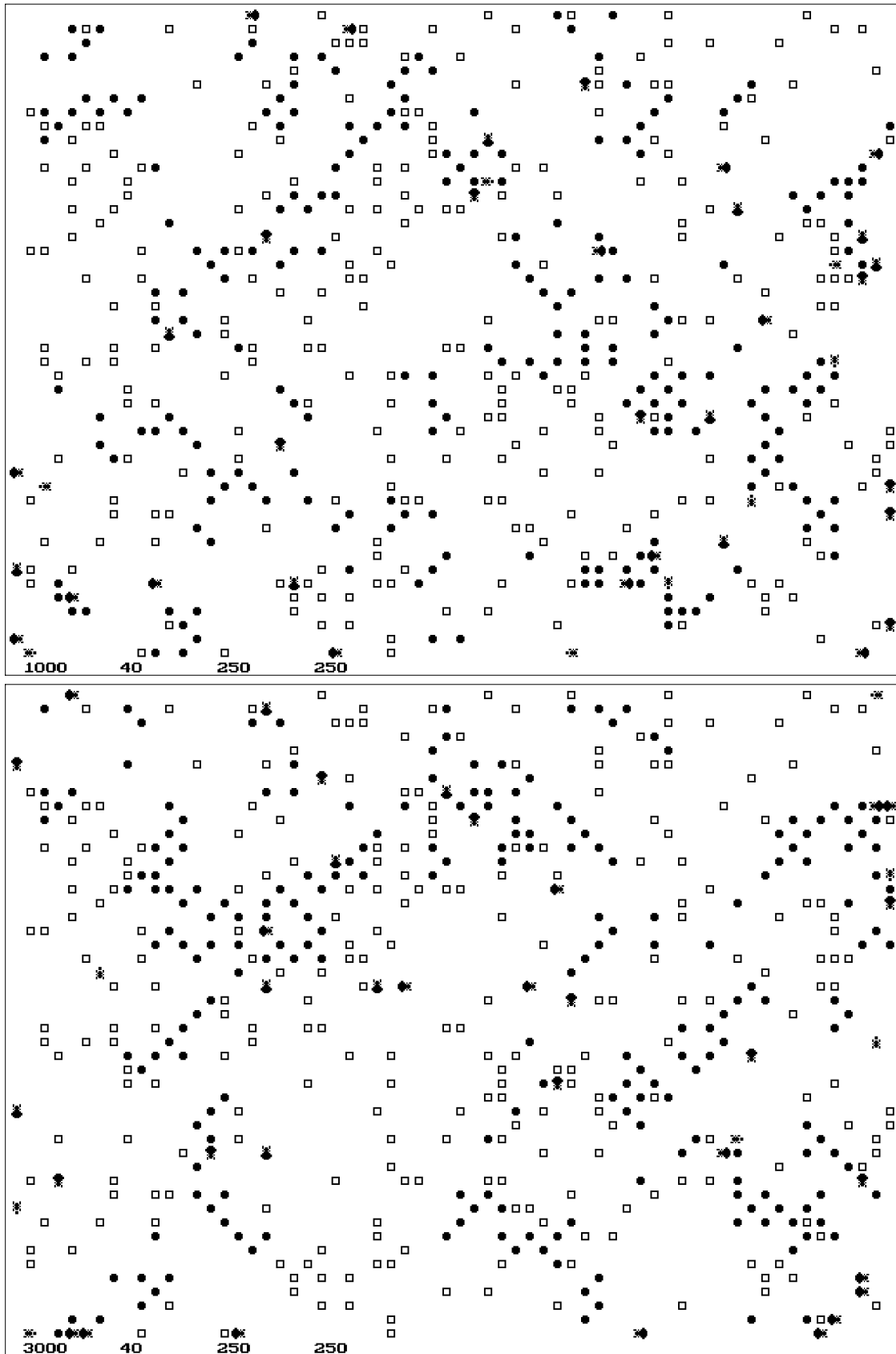
receptory: Straight, Left, Back, Right, HandStatus
efektory: Hand, Turn, Move
telo:      if HandStatus = free
              then
                if (Straight = rubbish) and (Left = rubbish) and
                  (Right = rubbish) and (Back = rubbish)
                  then Hand(takestraight); Turn(no!); Move(go!)
                  else Hand(no); Turn(no); Move(go)
              else
                if (Left = rubbish) and (Right = rubbish) and
                  (Back = nothing)
                  then Hand(putback); Turn(no!); Move(stop)
                  else Hand(no); Turn(no); Move(go)

```

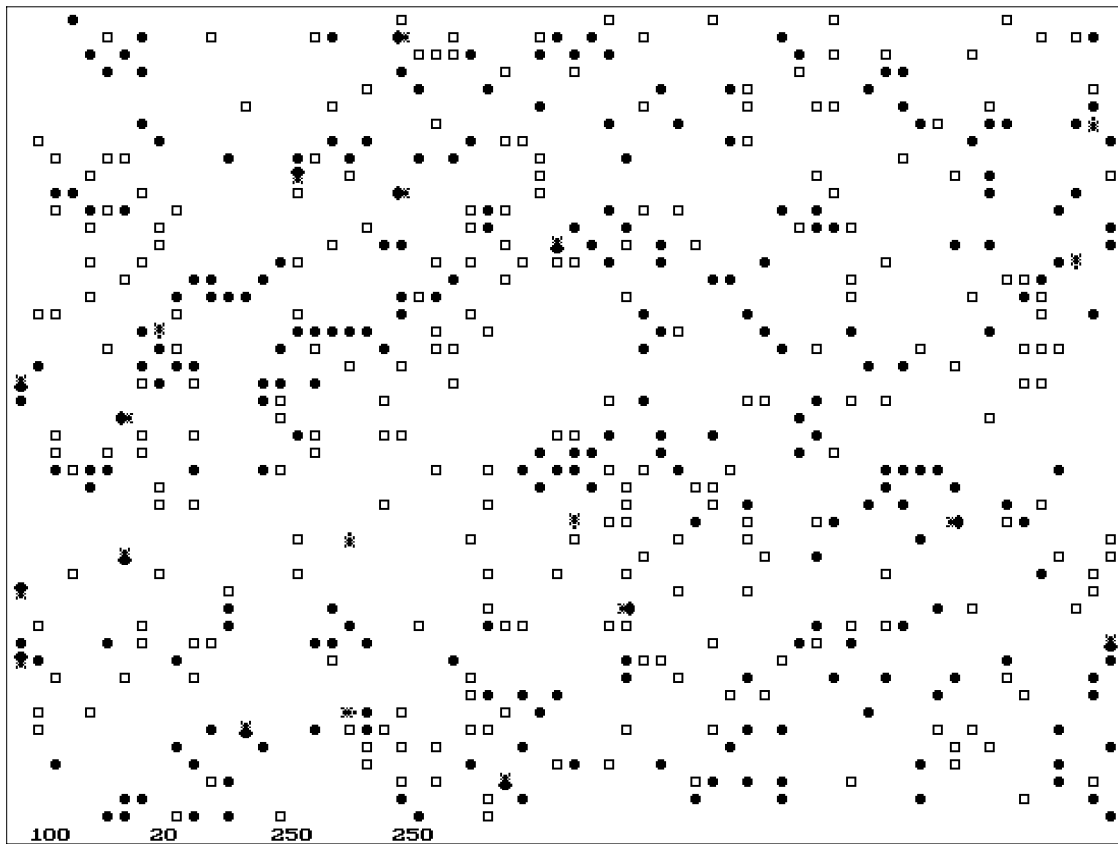
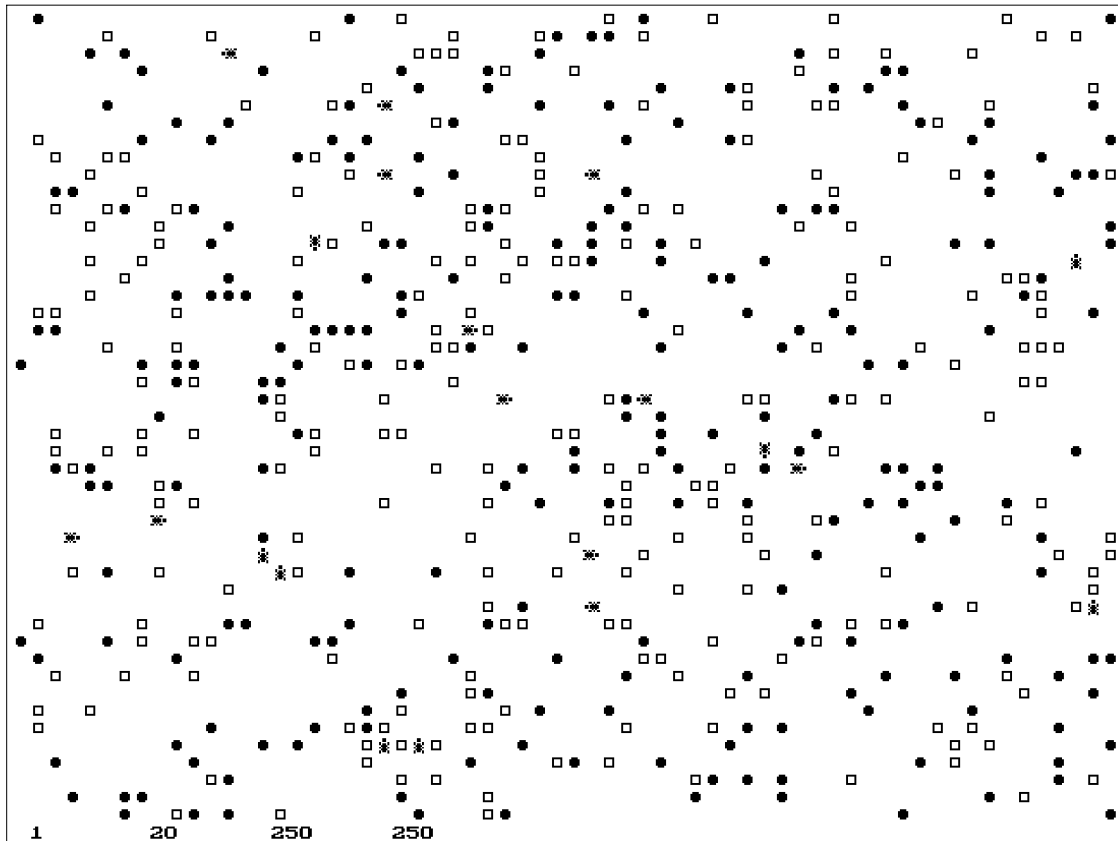
Ostatné časti agenta sú základné a sú popísané v kapitole 6.



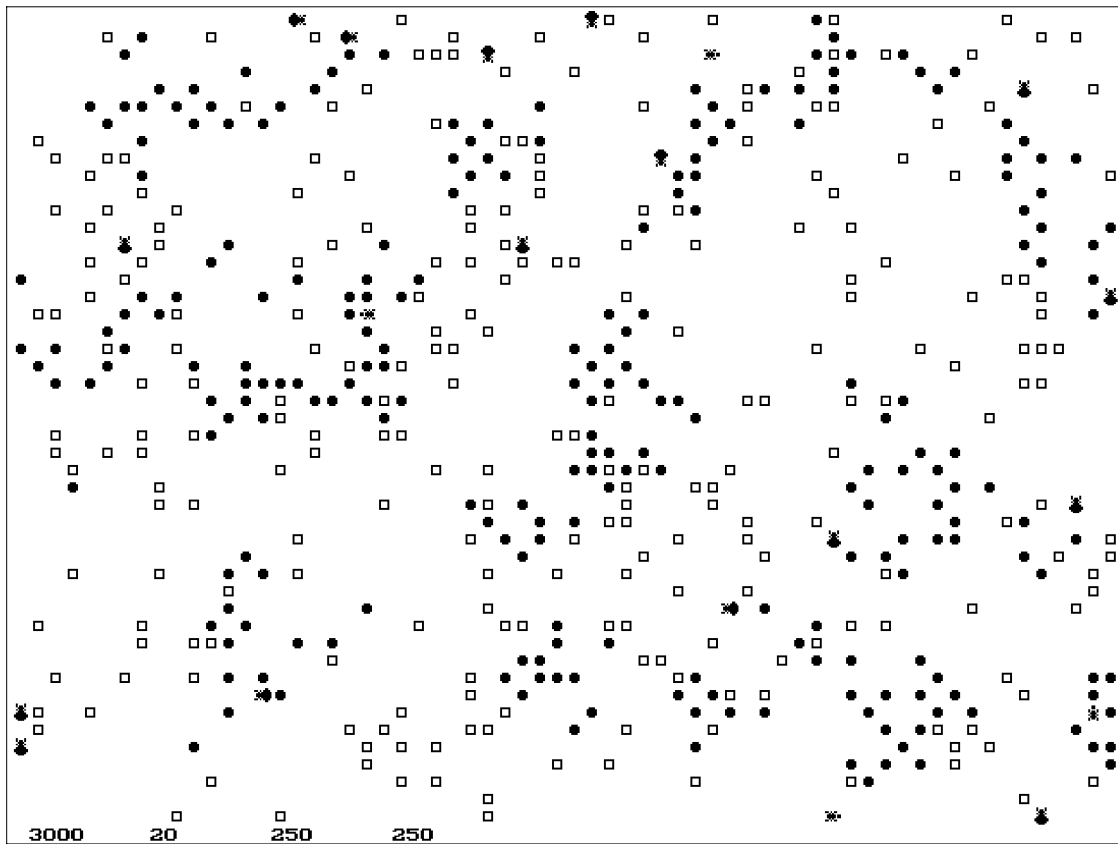
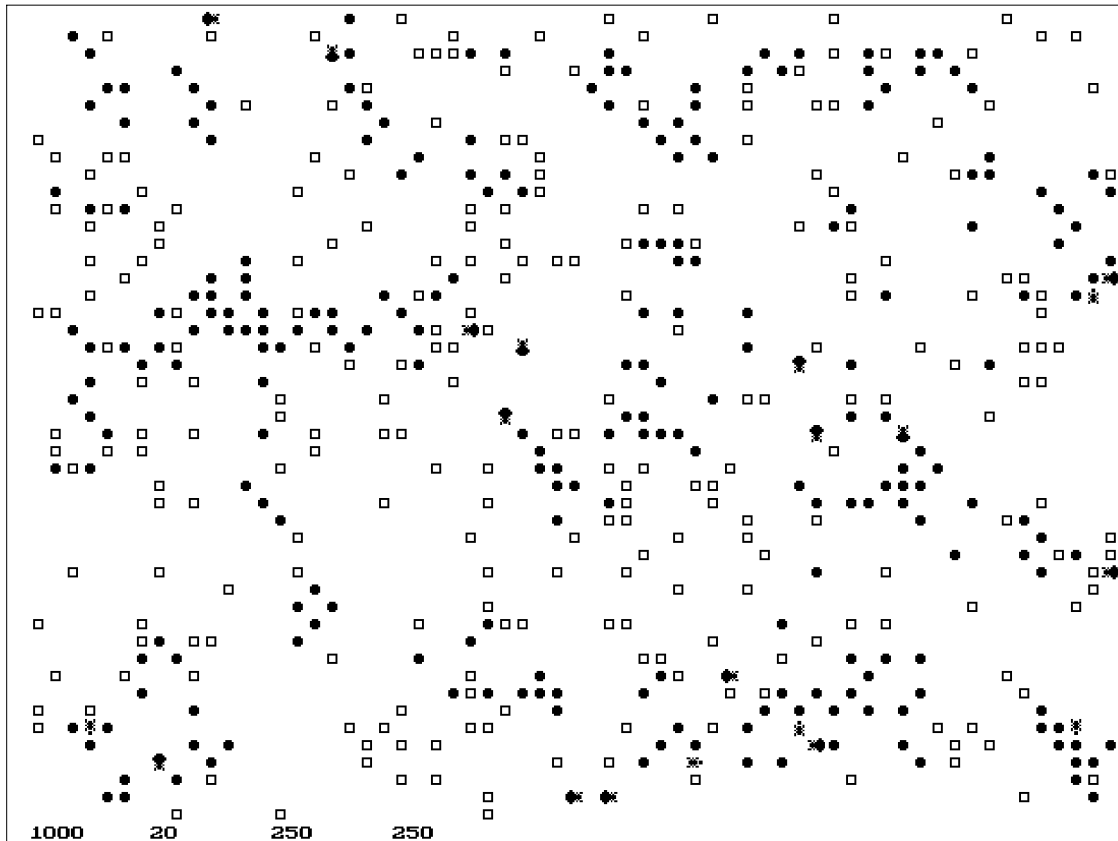
Obrázky 21a, 21b



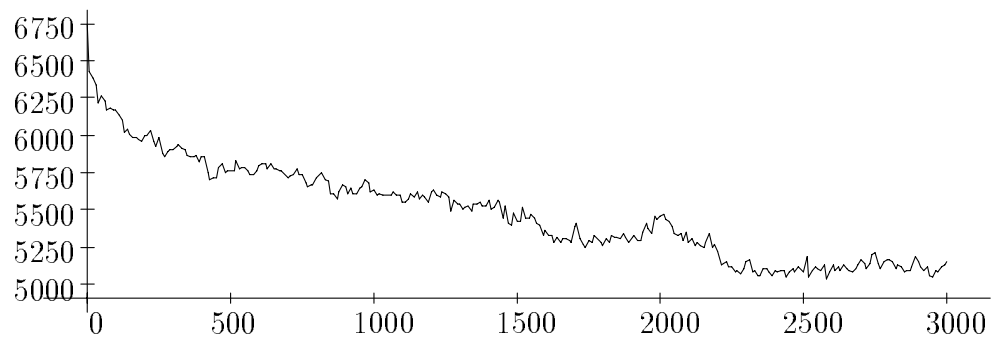
Obrázky 21c, 21d



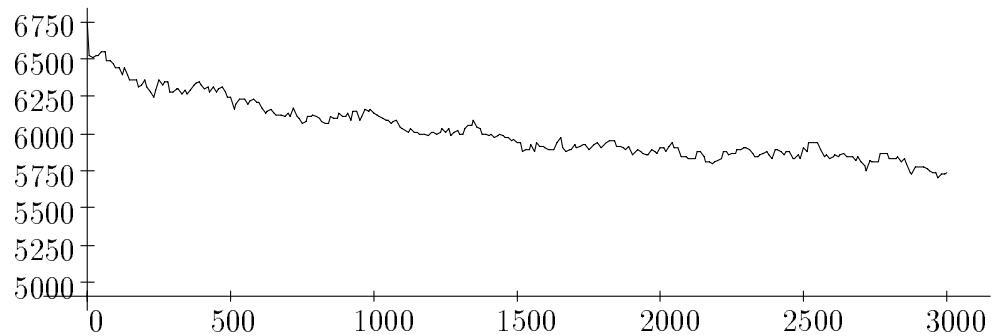
Obrázky 22a, 22b



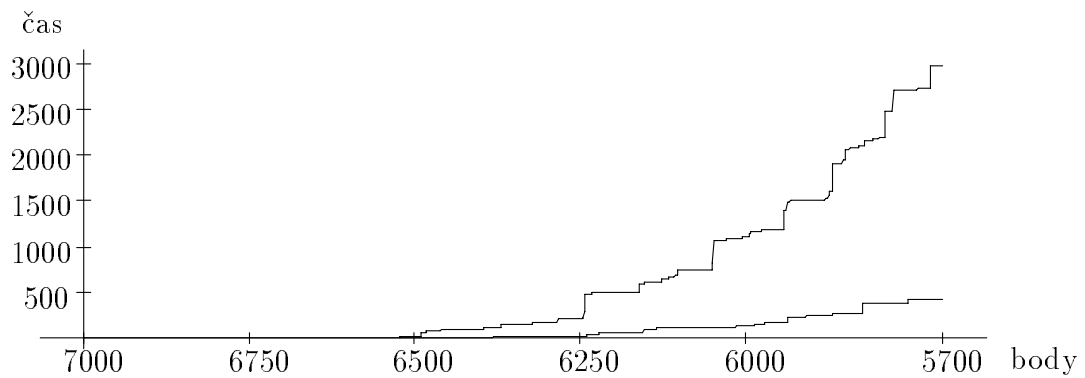
Obrázky 22c, 22d



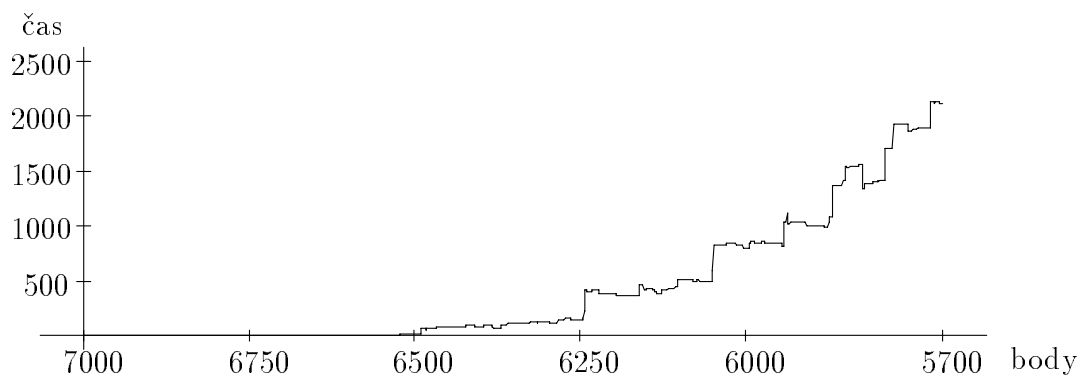
Obrázok 23



Obrázok 24



Obrázok 25



Obrázok 26

8.5 Výsledky modelovania vývoja kolónie

So zvolenou kolóniou sme pomocou systému **Mra** uskutočnili viacero modelovaní jej vývoja vo zvolenom prostredí. Pritom sme dospeli k nasledujúcim pozorovaniam:

- (1) Pri malom alebo veľkom pomere agentov a smetí je riešenie úlohy málo kvalitné. V prvom prípade je to očakávaný výsledok, lebo na málo agentov pripadá veľa smetí. V druhom prípade je to zavinené pravdepodobne tým, že vzniká príliš veľa názorov na to, kde má byť smetisko a kde čistinka, takže sa agenti nedohodnú. Extrémny prípad nastáva, keď je agentov viac ako smetí. Vtedy nastane okamih, keď sú všetky odpadky v rukách agentov a už v nich aj zostanú, pretože prostredie sa stane uniformným a agentom sa žiadne miesto v ňom nebude zdať vhodné na založenie smetiska.
- (2) Pri primeranom pomere agentov a smetí bude kolónia úlohu celkom pekne riešiť, čo znamená, že agenti sa dohodnú na tom, kde budú smetiská a kde čistinky. Ako to robia, keď spolu priamo nekomunikujú a navzájom na seba skoro nereagujú (iný agent je pre agenta iba polybujúca sa prekážka)? Robia to tak, že komunikujú cez prostredie. Pritom sa dvaja komunikujúci agenti nemusia nikdy stretnúť, stačí, keď jeden druhému podhodí do cesty odpadok. V modelovanej kolónii toto podhodenie odpadku predstavuje návrh na vybudovanie smetiska adresovaný všetkým agentom, ktorí okolo neho pôjdu. Ak k nemu prihodia ďalšie, znamená to, že agenti s vybudovaním smetiska na tomto mieste súhlasia, ak ho naopak zodvihnú, znamená to ich nesúhlas. Samozrejme potom je na vybudovanie smetiska na určitom mieste potrebný určitý počiatkový podnet. Skutočnosť, že smetiská pri vývoji kolónie vznikajú, nasvedčuje tomu, že v prostredí sú od začiatku akési potenciálne smetiská - miesta, ktoré lákajú agentov, aby na nich zhadzovali smeti.

Modelovania s meraním vyhodnocovacej funkcie sme uskutočnili pre prostredie obsahujúce 250 kameňov a 250 smetí, kolóniu zloženú zo 40-tich agentov a $1/2$ -kolóniu z 20-tich agentov. Jeden z vývojov kolónie je zobrazený na obrázkoch 21a-21d. Analogický vývoj $1/2$ -kolónie môžete sledovať na obrázkoch 22a-22d. (Na obrázkoch môžete vidieť agentov v dvoch vonkajších stavoch: nesúcich smet' a nenesúcich nič.)

Meraniami aproximované funkcie f_1 a f_2 vidíme na obrázkoch 23 a 24. Ďalší obrázok 25 zobrazuje naraz funkcie f_1^{-1} , f_2^{-1} . Krivka kompetencie vypočítaná podľa týchto funkcií je na obrázku 26. Na nej vidíme, že v okolí medze 5700 je hodnota cc výrazne kladná. Na základe kritéria kompetencie môžeme teda prehlásiť, že kolónia zbiera smeti **kompetentne**. Schopnosť agentov dohodnúť sa na tom, kde budú smetiská a kde čistinky rastie pre daný počet smetí do istej hranice s počtom agentov. V kolónii teda vzniká **emergentný efekt: dohoda o spôsobe plnenia kolektívneho cieľa**.

Čím je spôsobený tento efekt? Tento efekt je spôsobený komunikáciou uskutočňovanou cez prostredie, ktorej intenzita narastá s počtom agentov. Keď však táto

intenzita stúpne nad kritickú hodnotu, agenti sa nedohodnú. Dalo by sa povedať, že medzi nimi vypukne hádka. Správaniu agentov vrámci kolónie možno ľahko prisúdiť vlastnosti, na ktoré sme my ľudia takí pyšní.

8.6 Výsledky experimentu

Tento experiment ukázal, že prostredníctvom komunikácie cez prostredie môžu byť realizované aktivity, ktoré sa všeobecne považujú za realizovateľné iba pomocou priamej adresnej komunikácie. Experiment ďalej ukázal, že na to, aby prebiehala komunikácia sa nemusia komunikujúci agenti stretnúť a nemusia mať ani schopnosti naväzovať priame spojenie na diaľku. Ukázalo sa, že komunikácia cez prostredie napomáha riešeniu úlohy len pri primeranom počte komunikujúcich. Ďalej sa ukázalo, že každé prostredie obsahuje miesta potenciálne vyvolávajúce podnety pre určité aktivity. Zdá sa, že tieto podnety sú nutnou podmienkou tvorivosti, takže v uniformnom prostredí nie je možné dosiahnuť emergentné efekty.

Kapitola 9

Experiment: Stretnutie na námestí

9.1 Prostredie

Prostredie obsahuje jeden druh objektov – kamene (identifikačné číslo `stone`), ktoré sú v počiatočnej momentke náhodne umiestnené v priestore 64×47 .

9.2 Schopnosti agentov

Agenti sú schopní vnímať svoje bezprostredné okolie a rozpoznávať v ňom kamene a iných agentov. Agenti nie sú homogénni a to takým spôsobom, že možno na pohľad určiť, ktorým smerom sa agent pohybuje. Agenti budú mať teda prednú časť odlišiteľnú od zadnej časti. Prednú časť budeme nazývať hlavou. Agenti budú mať schopnosť rozpoznať u iného agenta, ktorý sa nachádza v jeho okolí, či je agent k nemu orientovaný hlavou (pohybuje sa smerom k nemu) alebo nie (pohybuje sa smerom od neho). Podobne ako v predchádzajúcich experimentoch agenti nedokážu kamene ani premiestňovať, ani ničiť.

Na označenie identifikačného signálu tohto druhu agentov budeme používať konštantu `agent`.

9.3 Úloha

Agenti sú v počiatočnej momentke rozptýlení v celom priestore. Úlohou kolónie, ktorá je z nich zložená, je vytvoriť čo najmenší počet stabilných skupín, v ktorých sú agenti v kontakte. Agenti teda budú vyvíjať snahu stretnúť sa a držať sa spolu.

Vyhodnocovaciu funkciu definujeme ako podiel počtu skupín agentov v danej momentke a počtu agentov, vynásobený stami. Na presné zadefinovanie pojmu skupina potrebujeme najprv pojem kontakt dvoch agentov. Budeme hovoriť, že dvaja agenti a_1, a_2 sú v kontakte, keď sú miesta $miesto(a_1), miesto(a_2)$ susedné alebo keď existuje agent a_3 , ktorý je s agentami a_1, a_2 v kontakte (miesta (r_1, c_1) a (r_2, c_2) sú susedné vtedy, keď existuje smer $(d_r, d_c) \in S_{\mathbf{Mra}}$, taký, že $r_1 + d_r = r_2$ a $c_1 + d_c = c_2$). Kontakt agentov je teda binárna relácia, ktorá je ekvivalenciou a ktorú budeme označovať \sim . Skupinou agentov potom definujeme ako množinu agentov, v ktorej sú každý dvaja agenti v kontakte. Vyhodnocovaciu funkciu teda môžeme presne definovať ako

$$e(s) = \frac{\left| (s \cap \mathcal{A}) / \sim \right|}{|s \cap \mathcal{A}|} \cdot 100\%$$

Vidíme, že vyhodnocovacia funkcia vyjadruje intenzitu agregácie agentov v percentách. Čím menšia je hodnota vyhodnocovacej funkcie, tým sú agenti viac agregovaní.

Za medzu úlohy zvolíme malú ale nenulovú intenzitu agregácie $\nu = 10\%$.

9.4 Riešenie: zloženie agenta a kolónie

Kolónia navrhnutá na riešenie bude obsahovať agentov jediného druhu. Agent bude pozostávať:

- z receptorov `Straight, Left, Right, Back, HeadStraight, HeadLeft, HeadRight, HeadBack, MoveStatus, Rnd`
- z efektorov `Move, Turn` a
- z behaviorálnych modulov `Moving3, Avoidance, Joining, Retreating`.

Receptory `HeadStraight, HeadLeft, HeadRight` a `HeadBack` sprostredkujú agentovi informáciu o orientácii iného agenta, ktorý sa nachádza v jeho bezprostrednom okolí. Do behaviorálnych modulov vysielajú tri druhy signálov: `noagent, yes` a `no`. Prvý signál zodpovedá skutočnosti, že v danom smere sa v bezprostrednom okolí agenta iný agent nenachádza, ostatné signály zodpovedajú prítomnosti iného agenta v danom smere a vzťahujú sa na jeho relatívnu orientáciu.

`HeadStraight` :

$$(s, (y, x), (d_y, d_x)) \mapsto \begin{cases} \text{yes} & \exists o \in s : \begin{cases} miesto(o) = (y + d_y, x + d_x) \\ smer(o) = (-d_y, -d_x) \end{cases} \\ \text{no} & \exists o \in s : \begin{cases} miesto(o) = (y + d_y, x + d_x) \\ smer(o) \neq (-d_y, -d_x) \end{cases} \\ \text{noagent} & \text{inak} \end{cases}$$

Receptory `HeadLeft`, `HeadRight` a `HeadBack` sú definované podobne.

Modul `Moving3` má podobnú úlohu ako moduly `Moving` a `Moving2` v predchádzajúcich experimentoch, len s tým rozdielom, že nezabezpečuje nepokoj agenta. Ten v tomto experimente nie je žiadúci, nakoľko sa agenti majú držať spolu v skupinách.

Modul `Joining` zabezpečuje pripájanie sa agenta k už existujúcim skupinám.

Modul `Retreating` zabezpečuje útek agenta pred iným agentom, ktorý doňho vrazil. Tento útek často spôsobí spojenie dvoch skupín agentov, ale jeho hlavný význam spočíva v tom, že umožní plynulý pohyb celej skupiny agentov.

Joining

receptory: `Straight`, `Left`, `Right`, `HeadLeft`, `MoveStatus`, `Rnd`

efektory: `Turn`

```

telo:      if MoveStatus = ok
              then
                if Straight = agent
                  then Turn(no!)
                else
                  if (Left = agent) and (Straight = agent)
                    then
                      if HeadLeft = yes
                        then Turn(right!)
                      else Turn(left!)
                    else
                      if Left = agent
                        then
                          if Rnd > 1/60
                            then Turn(left!)
                          else Turn(no!)
                      else
                        if Right = agent
                          then
                            if Rnd > 1/60
                              then Turn(right!)
                              then Turn(no!)
                            else Turn(no)
                      else Turn(no)

```

Moving3

*receptory:**efektory:* Move*telo:* Move(go)

Retreating

receptory: Straight, Left, Back, Right, HeadStraight, MoveStatus*efektory:* Turn

```

telo:    if (MoveStatus = collision) and (Straight = agent)
           then
             if Back = nothing
               then Turn(back!)
             else
               if Left = nothing
                 then Turn(left!)
               else
                 if Right = nothing
                   then Turn(right!)
                 else Turn(no!)
           else Turn(no)

```

Ostatné časti agenta sú základné a sú popísané v kapitole 6.

9.5 Výsledky modelovania vývoja kolónie

So zvolenou kolóniou sme pomocou systému *Mra* uskutočnili viacero modelovaní jej vývoja vo zvolenom prostredí. Pritom sme dospeli k nasledujúcim pozorovaniam:

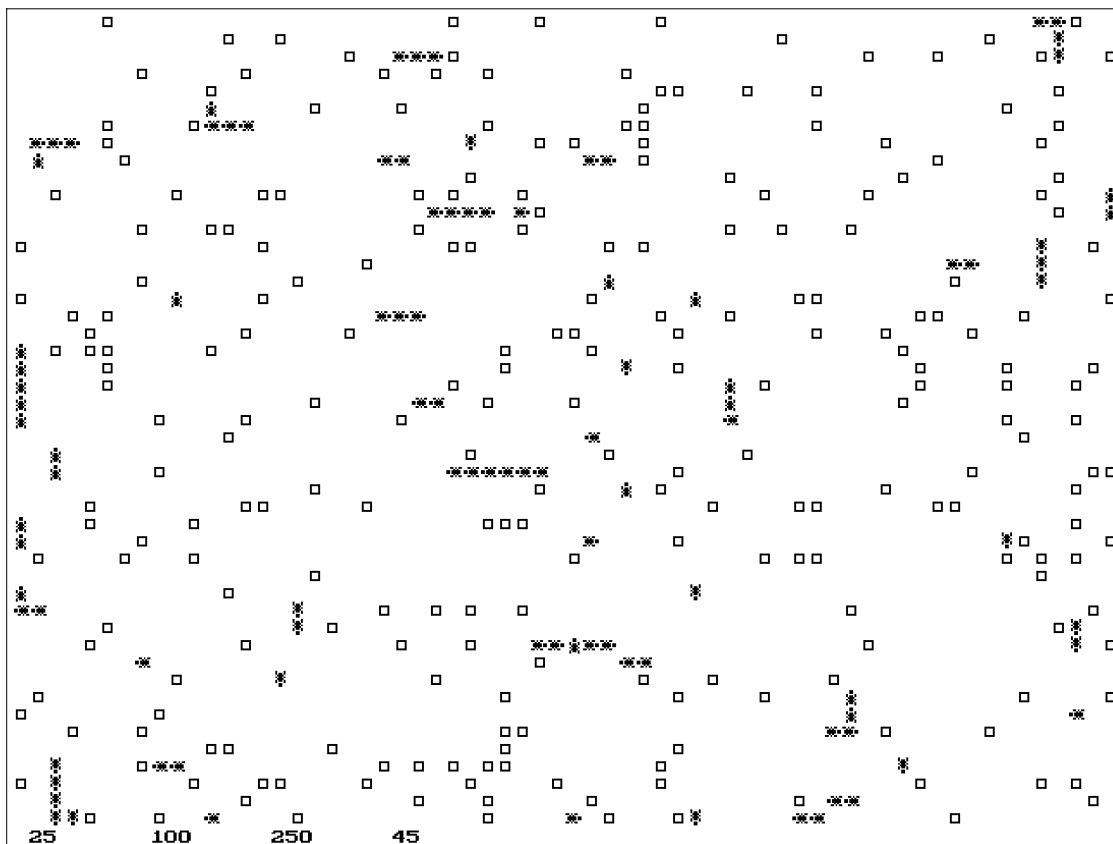
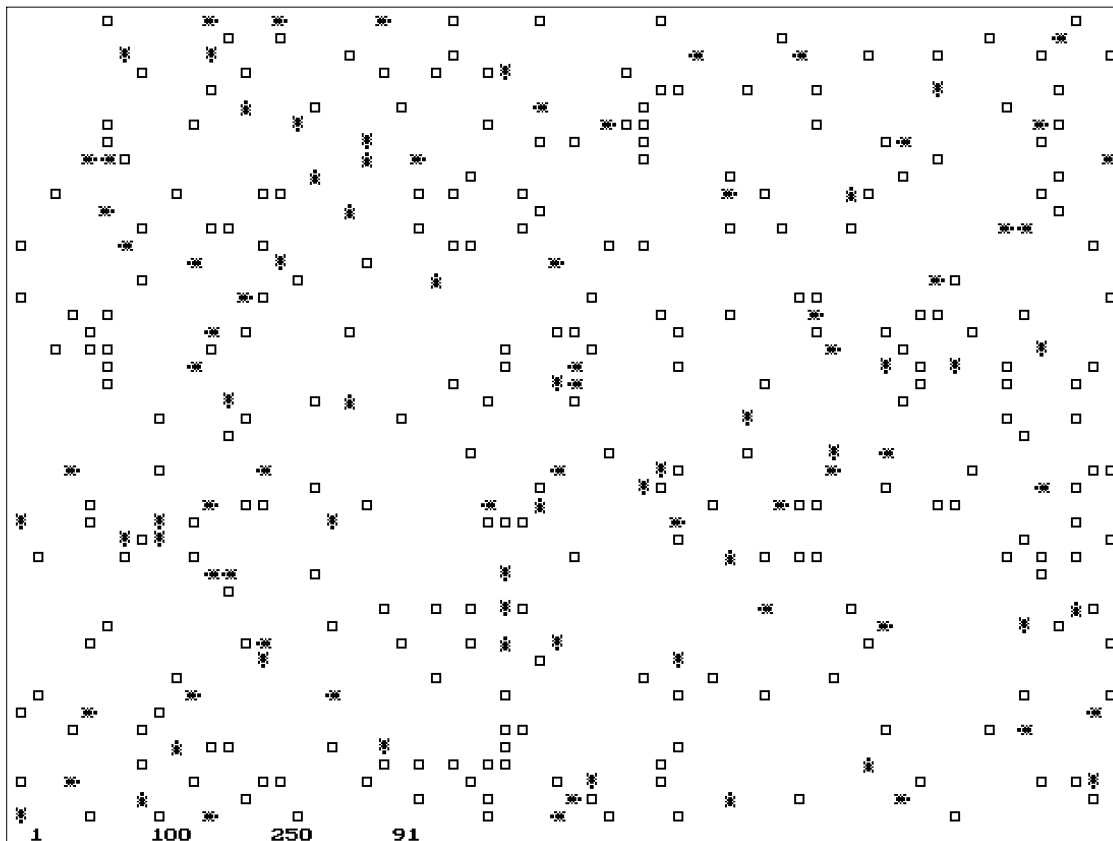
- (1) Agenti sa vo vývoji kolónie spájajú do väčších skupín, pričom každá skupina má tendenciu stať sa zástupom. Rýchlosť vytvárania týchto zástupov rastie s počtom agentov. Keď sa skupina sformuje do tvaru zástupu, svedčí to o tom, že agenti sa už dohodli kto bude ich vodcom a teraz sledujú jeho pohyby.
- (2) Pri zrážke dvoch zástupov môže prísť k spojeniu oboch skupín, úplnému rozbitiu jednej zo skupín, alebo spojeniu jednej skupiny s časťou druhej a osamostatneniu zvyšku druhej skupiny.

- (3) Zástup sa stále pohybuje. Pri pohybe sa však po čase dostane na miesto, kde sa usadí. Usadenie sa je často sprevádzané zrážkou s inou skupinou, ale nastáva i samostatne.
- (4) Pri usadení sa časť skupiny vytvorí „bludný“ kruh, v ktorom sa agenti pohybujú, z ktorého vychádzajú vetvy stojacich agentov. Skupina sa teda nehýbe z miesta.
- (5) Pokiaľ do usadenej skupiny pribúdajú noví členovia, skupina má tendenciu zostať usadená. Ak však dlho nepribudol žiaden člen, skupina má tendenciu rozmotáť bludný kruh a dať sa znovu do pohybu ako zástup.
- (6) Usadená skupina sa môže dať znovu do pohybu i vplyvom inej skupiny, ktorá prešla okolo nej v tesnej blízkosti a stiahla časť, alebo prípadne celú usadenú skupinu so sebou.
- (7) Existujú však i usadené skupiny, ktoré aj napriek nepribúdaniu nových členov ostávajú usadené. Tie vznikajú na špeciálnych miestach prostredia, ktoré budeme nazývať **námestia**.
- (8) Po určitom čase sa rozdelenie agentov do malého počtu skupín usadených na námestiach ustáli. Nie každé potenciálne námestie bude obsadené nejakou skupinou. Počet skupín bude záležať ani nie od prostredia ale od počtu agentov: čím väčší počet agentov, tým viac skupín. S intenzitou agregácie to však bude presne opačne.

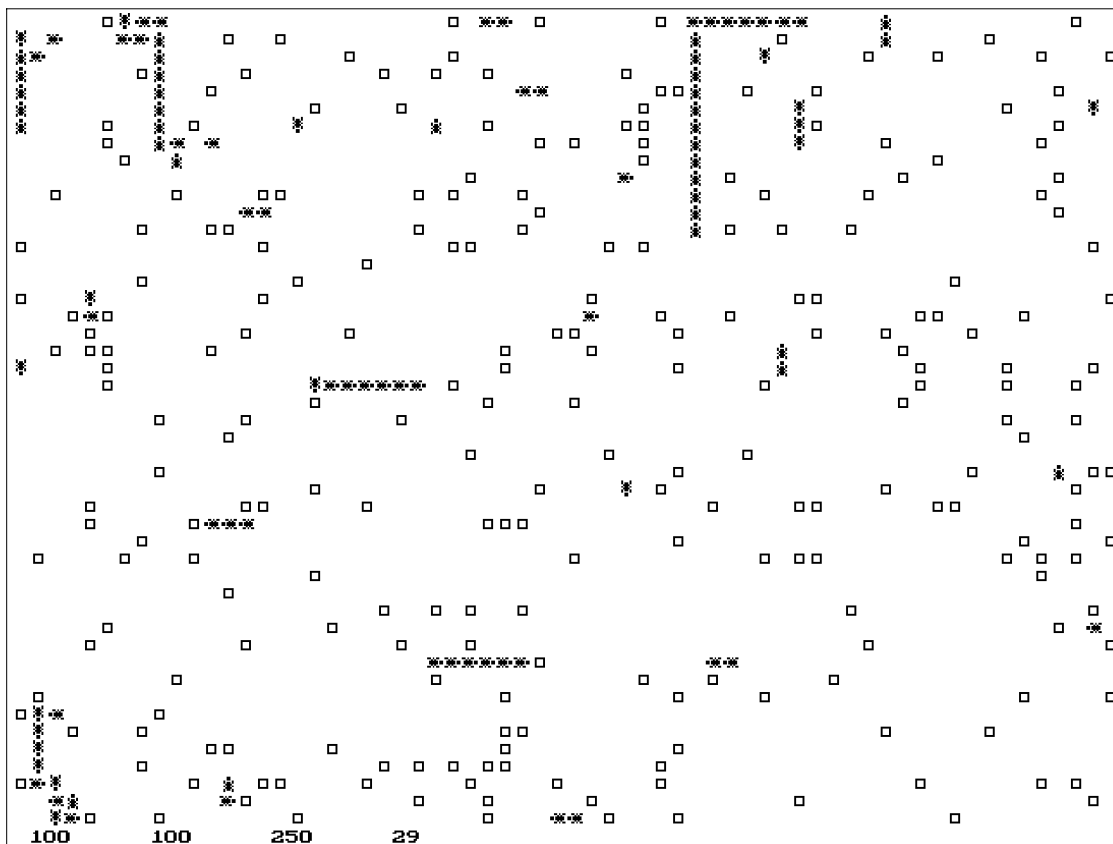
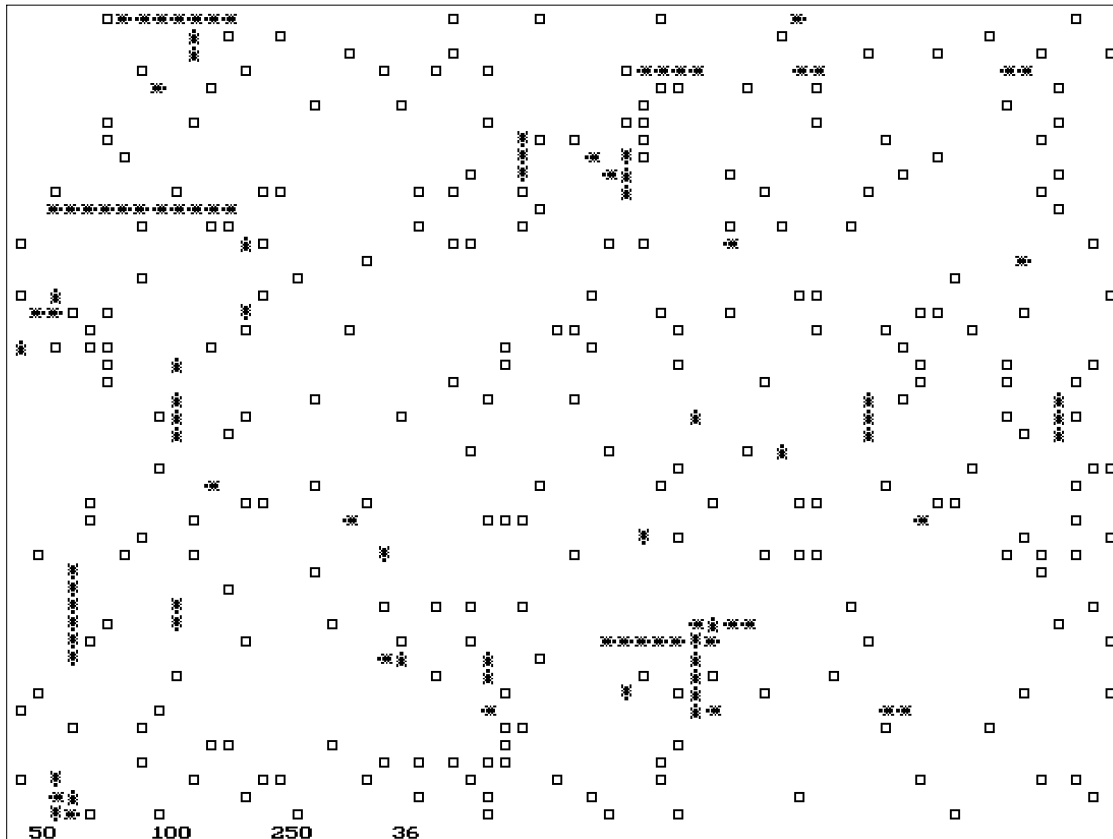
Modelovania s meraním vyhodnocovacej funkcie sme uskutočnili pre prostredie obsahujúce 250 kameňov, kolóniu zloženú zo 100 agentov a $1/2$ -kolóniu z 50-tich agentov. Jeden z vývojov kolónie je zobrazený na obrázkoch 27a-27f. Analogický vývoj $1/2$ -kolónie môžeme sledovať na obrázkoch 28a-28h.

Keďže priemery výsledkov meraní jednotlivých modelovaní môžu v tomto prípade výrazne vylepšovať riešenie úlohy, uvádzame ukážku priebehu vyhodnocovacej funkcie piatich meraní pre kolóniu (obrázok 28) a $1/2$ -kolóniu (obrázok 27). Tieto výsledky skutočne potvrdzujú, že sa rozdelenie agentov do skupín ustáli nielen v priemere, ale aj pri všetkých modelovaniach (s prostredím poskytujúcim námestia).

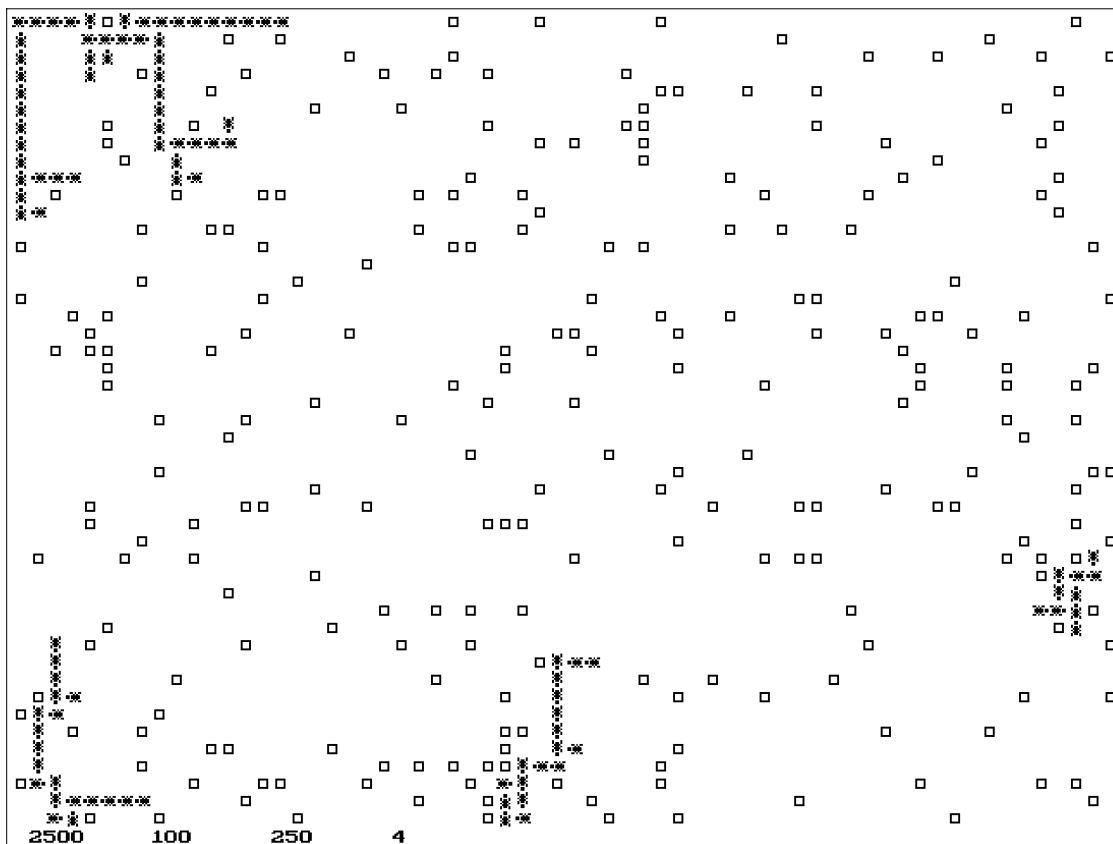
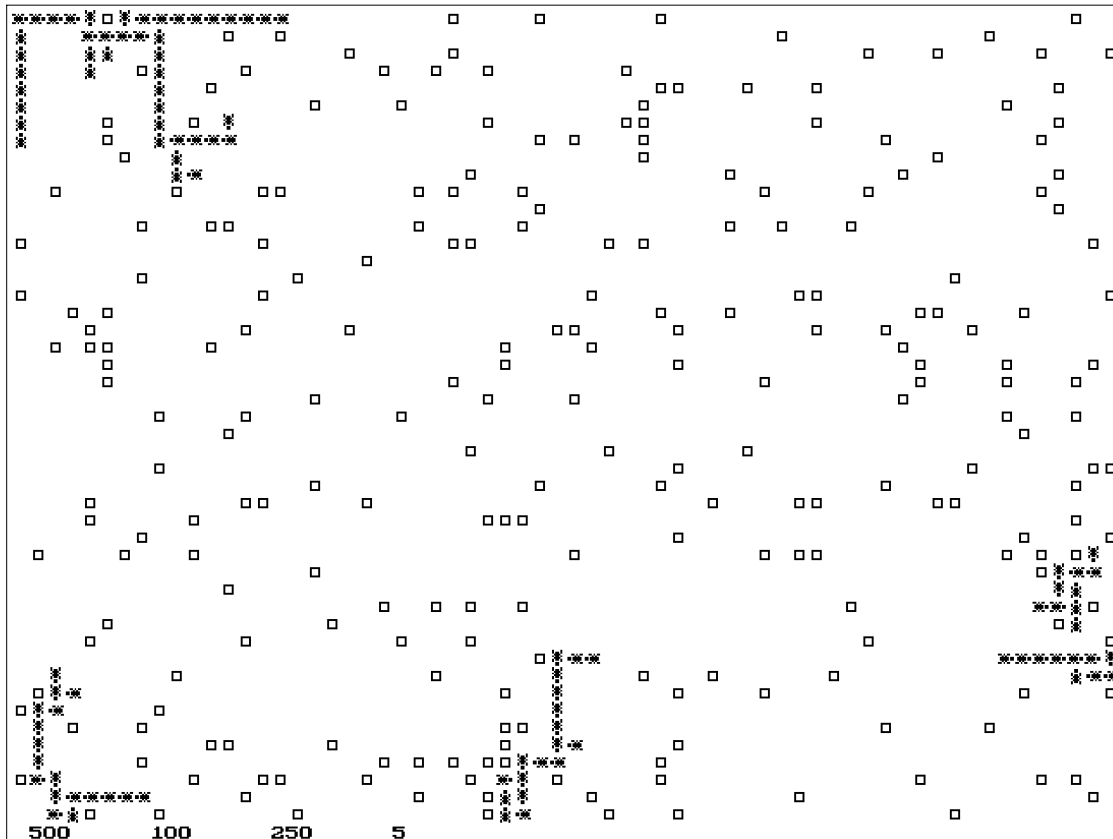
Meraniami aproximované funkcie f_1 a f_2 vidíme na obrázkoch 31 a 32. Ďalší obrázok 33 zobrazuje naraz funkcie f_1^{-1} , f_2^{-1} . Krivka kompetencie vypočítaná podľa týchto funkcií je na obrázku 34. Na nej vidíme, že v okolí medze 10% je hodnota cc výrazne kladná. Na základe kritéria kompetencie môžeme teda prehlásiť, že kolónia vytvára skupiny **kompetentne**. Intenzita agregácie rastie s počtom agentov. V kolónii vzniká **emergentný efekt: ustálenie sa vývoja z vlastnej iniciatívy**. Tu je skutočne namieste preštudovať si jednotlivé moduly a presvedčiť sa, že ustálenie sa vývoja v nich nie je naprogramované, naopak, naprogramované je, aby sa agenti neustále hýbali.



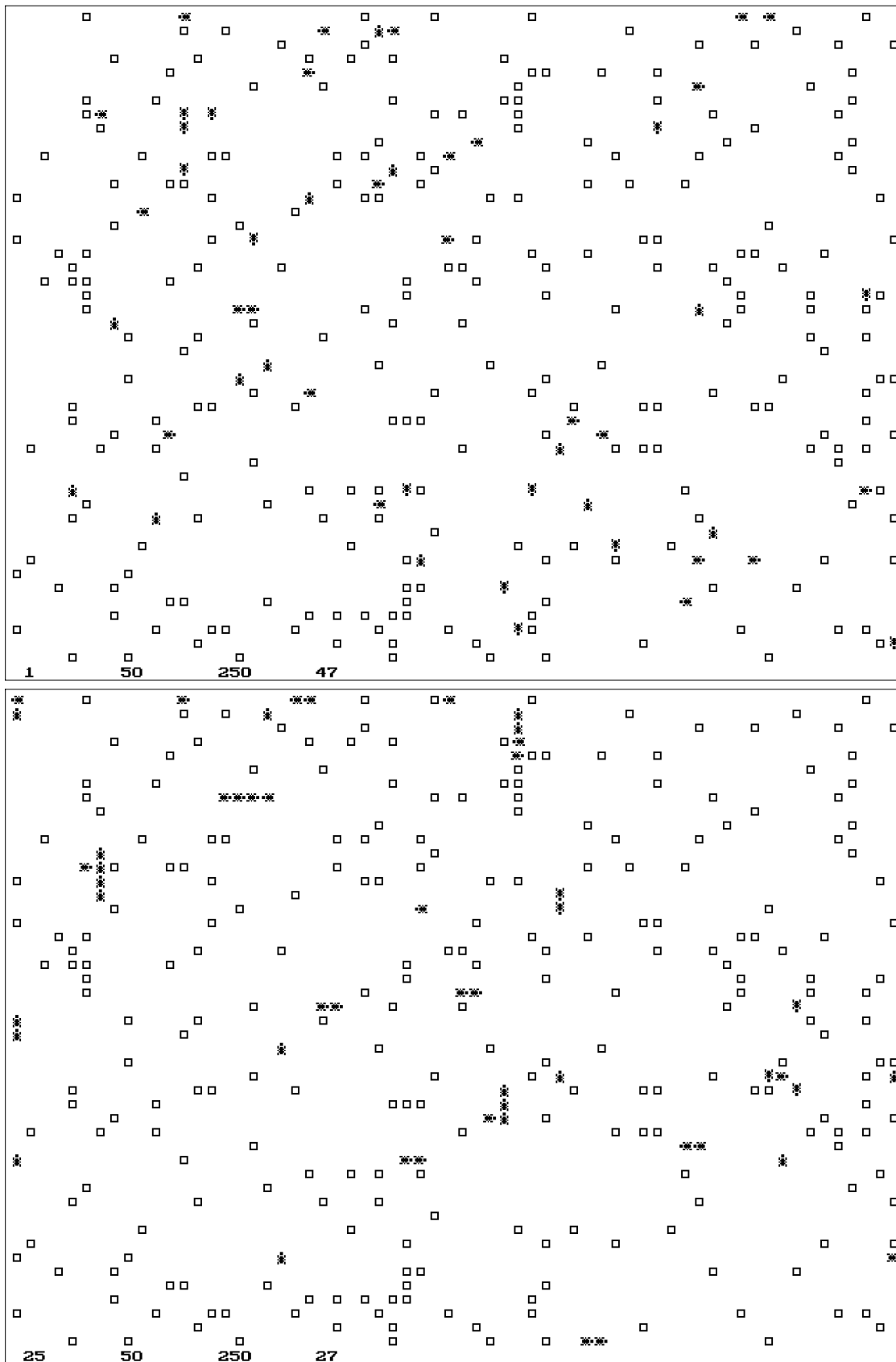
Obrázky 27a, 27b



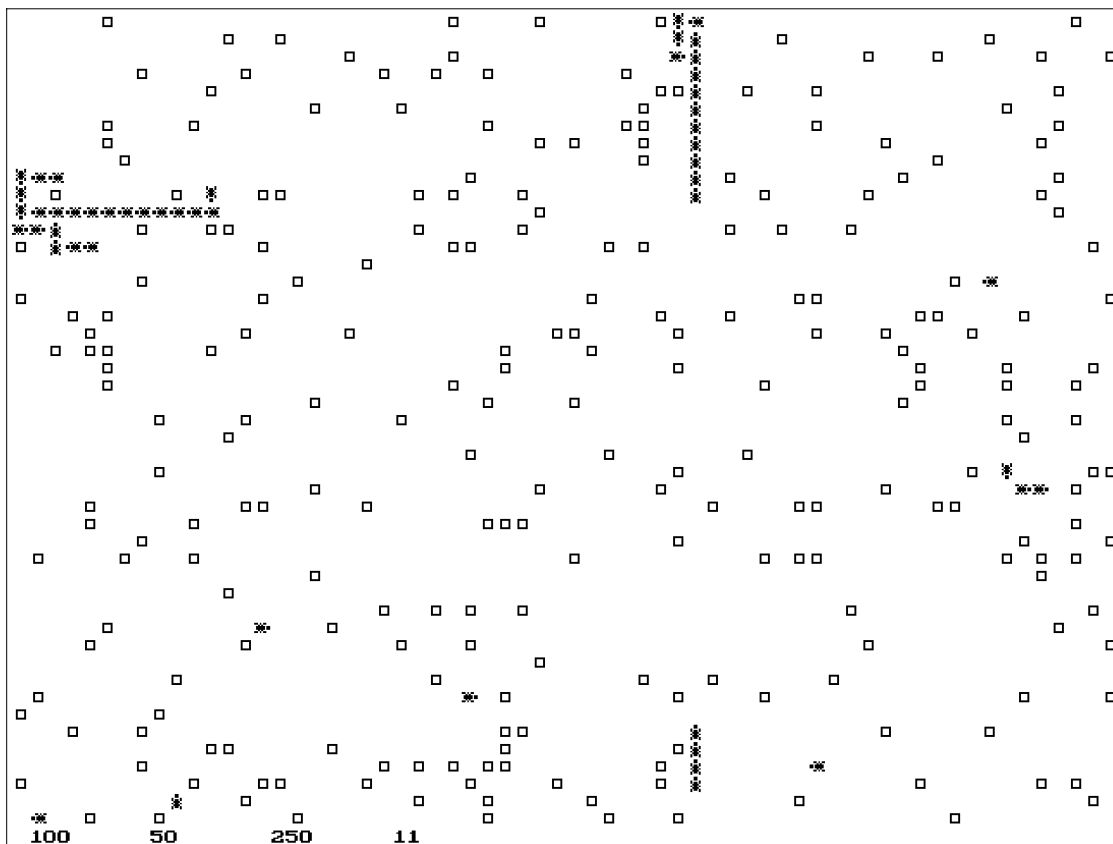
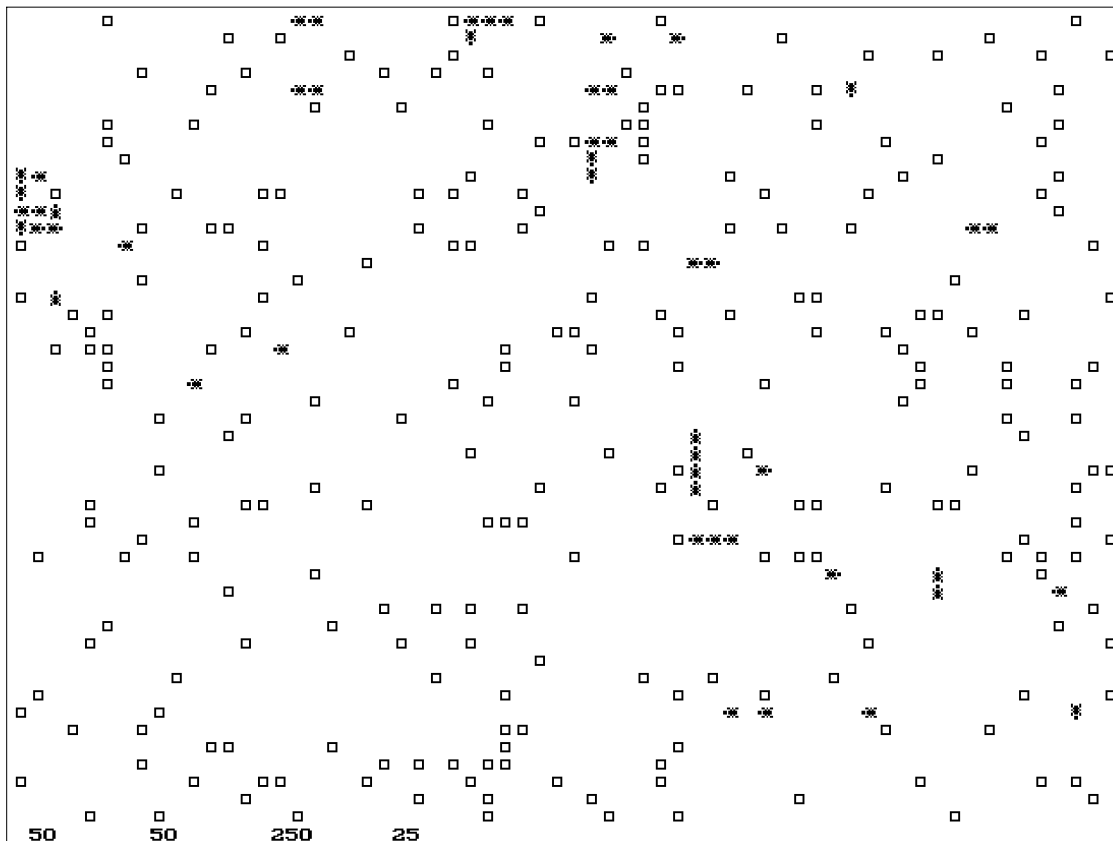
Obrázky 27c, 27d



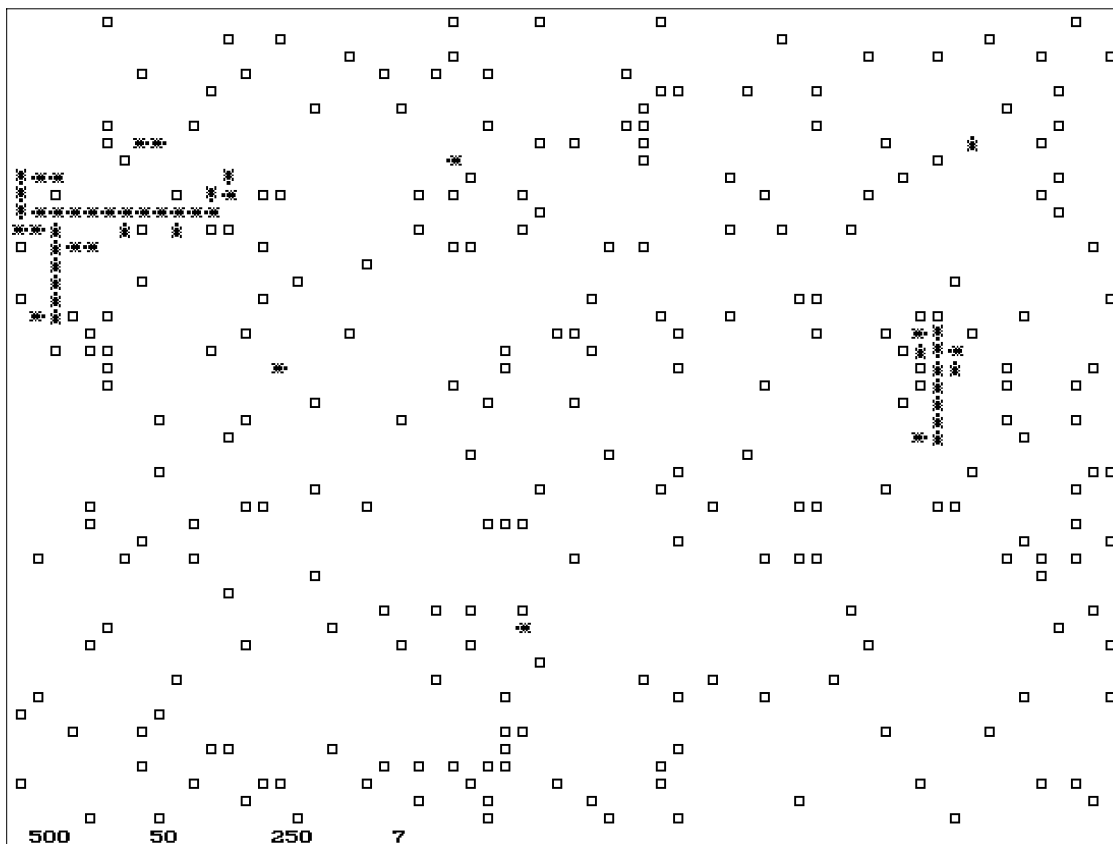
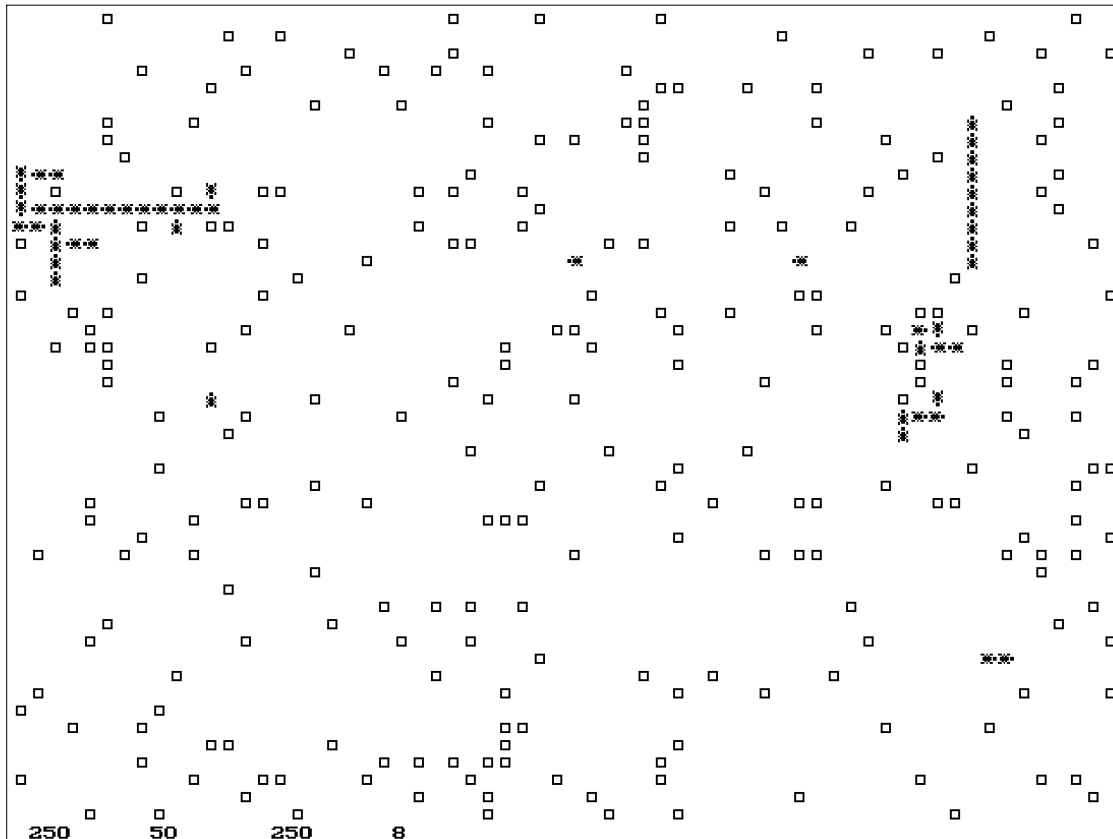
Obrázky 27e, 27f



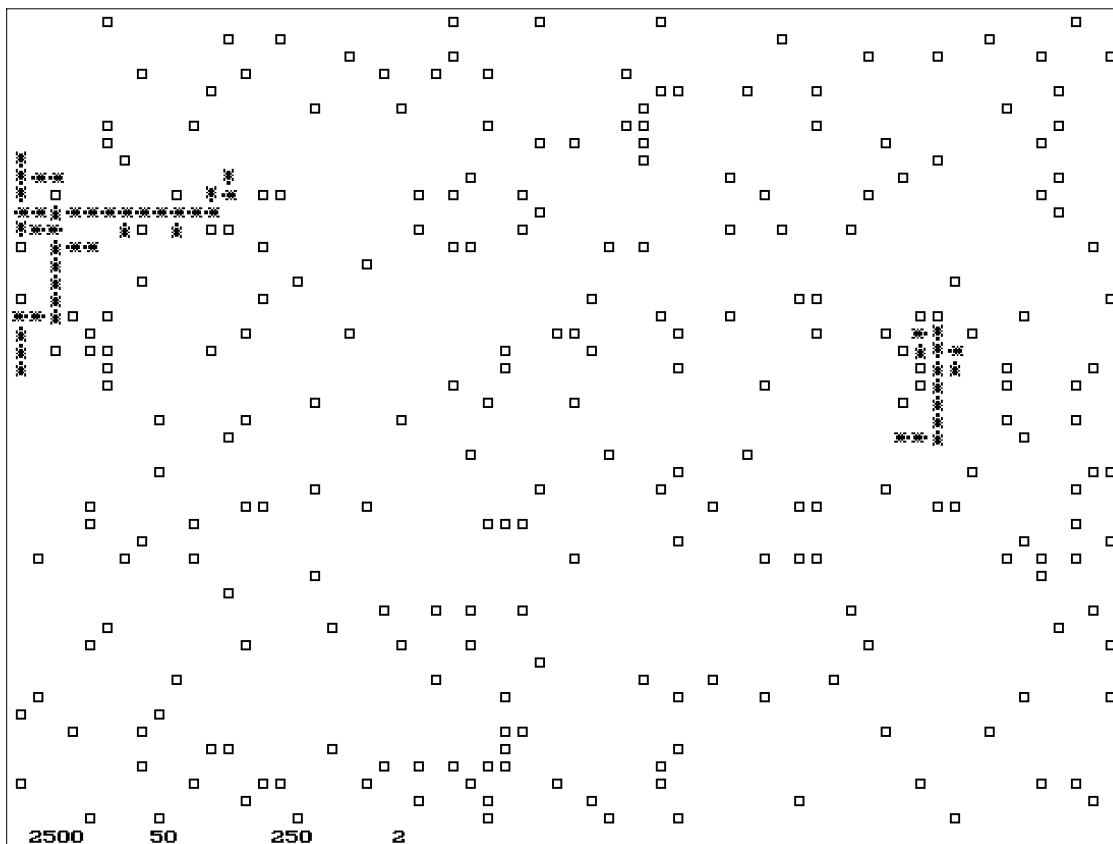
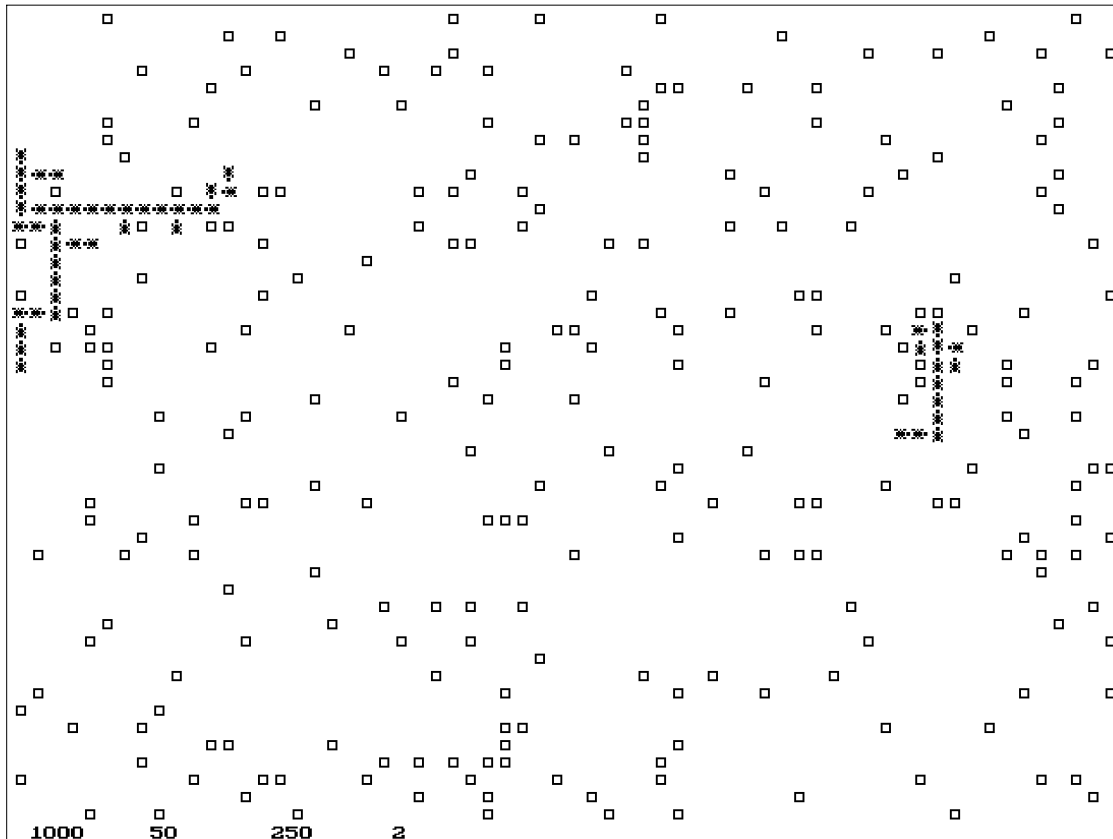
Obrázky 28a, 28b



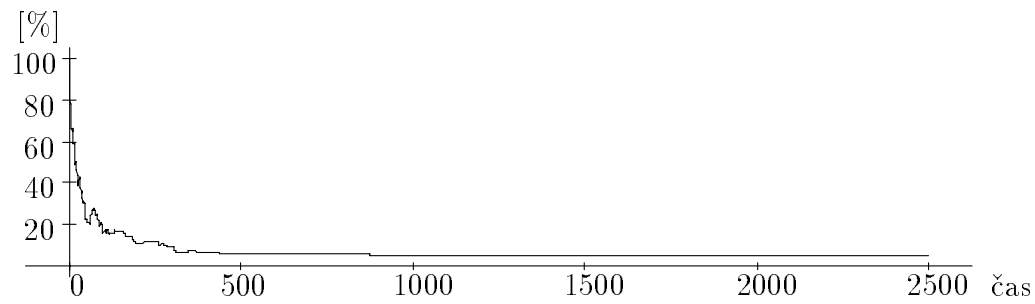
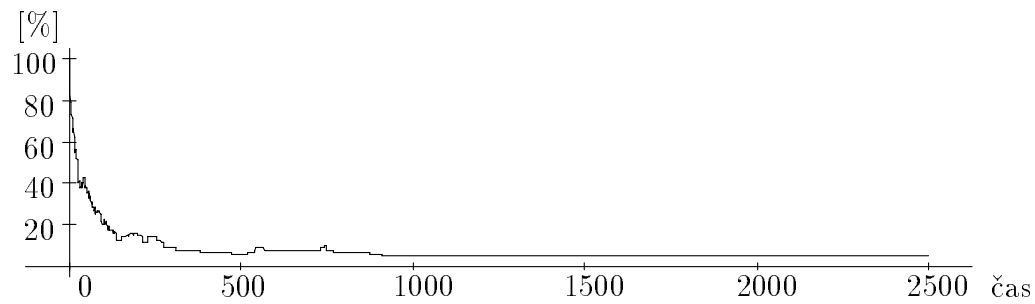
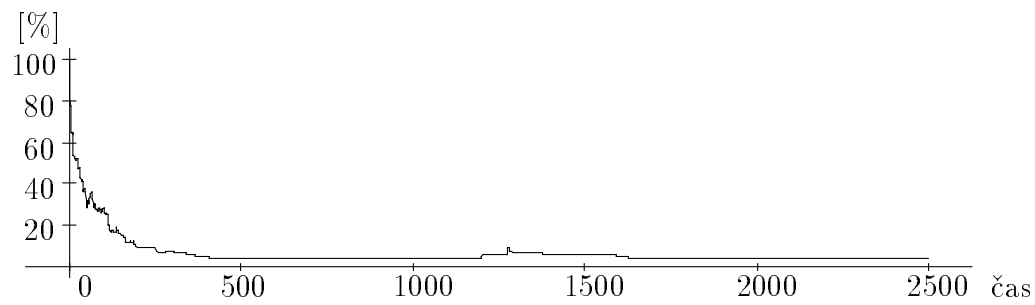
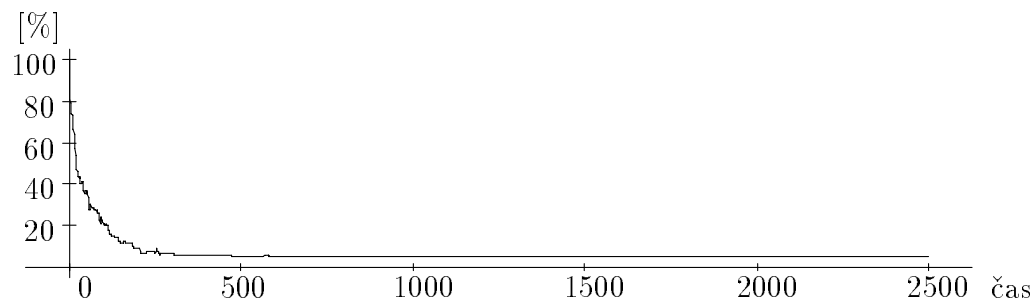
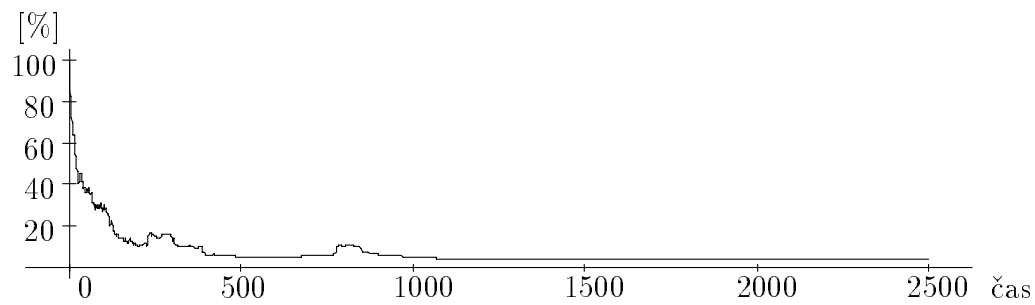
Obrázky 28c, 28d



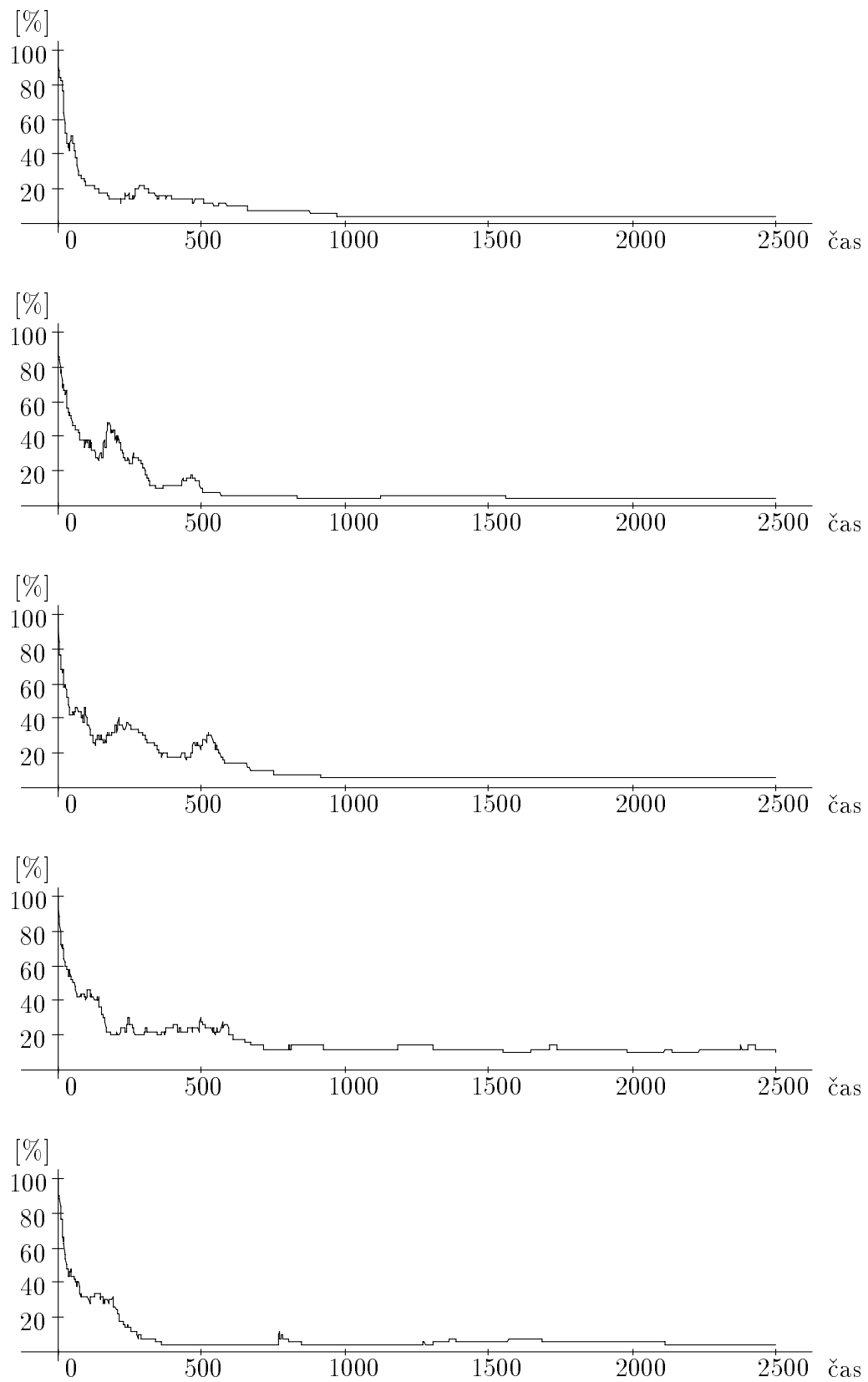
Obrázky 28e, 28f



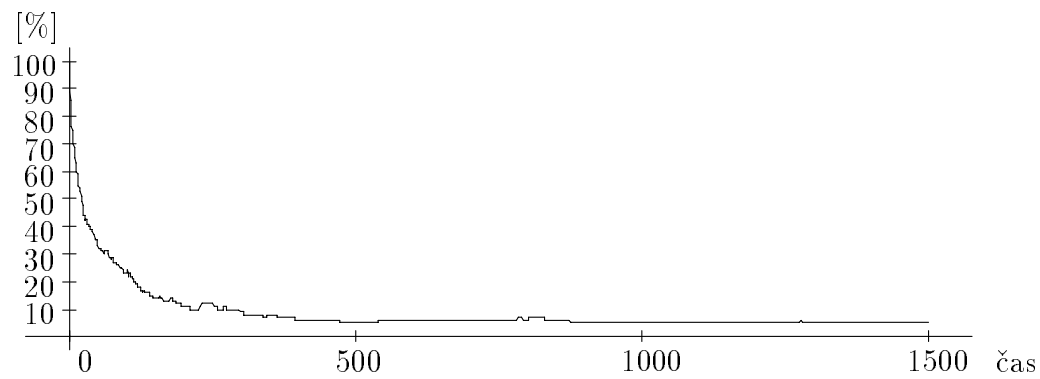
Obrázky 28g, 28h



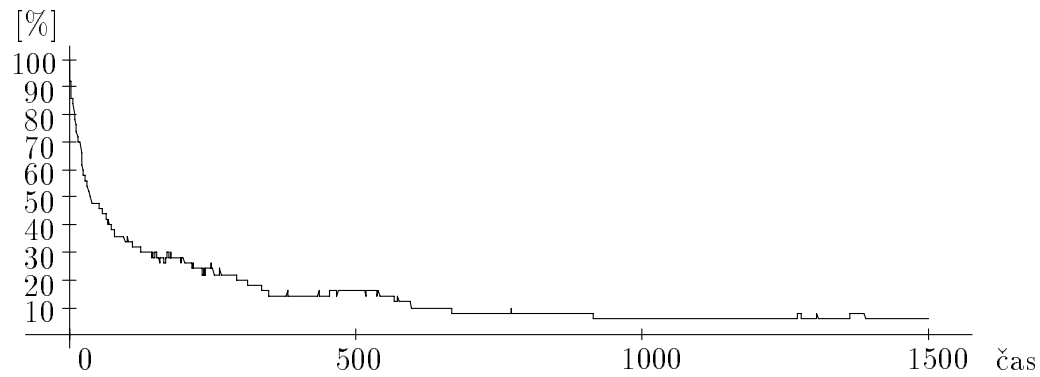
Obrázok 29



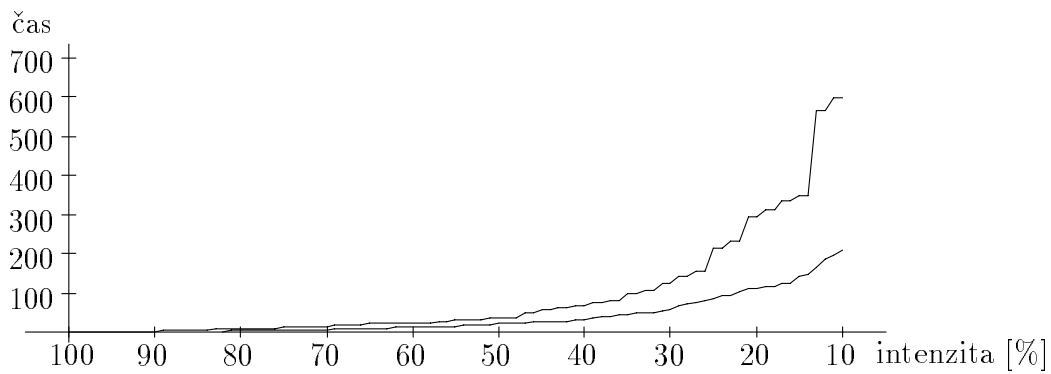
Obrázok 30



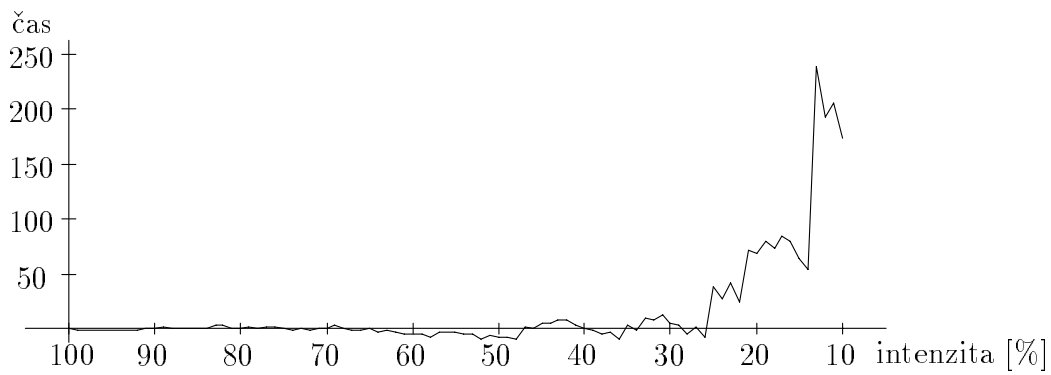
Obrázok 31



Obrázok 32



Obrázok 33



Obrázok 34

Čím je spôsobený tento efekt? Tento efekt je spôsobený pravdepodobne nejakým „defektom“ v naprogramovaní modulov, ktorý sa s určitou pravdepodobnosťou prejavuje na špeciálnych miestach prostredia. Tu však treba upozorniť na to, že vplyv tohto „defektu“ nemá vždy trvalé následky, čo nie je charakteristické pre defekty zatiaľ skonštruovaných strojov, avšak čo je oveľa charakteristickejšie pre živé tvory.

Veľmi zaujímavý efekt bolo možné pozorovať na skupine, ktorá sa bez prispenia inej skupiny usadila a potom sa opäť z vlastnej iniciatívy dala do pohybu. Akoby prekonala dočasné pôsobenie spomínaného „defektu“ a vrátila sa k pôvodnému režimu aktivít.

9.6 Výsledky experimentu

Tento experiment ukázal, že kolónia môže mať tendenciu ustáliť svoj vývoj, pričom táto tendencia vyplýva len z jej vnútornej povahy a povahy prostredia v ktorom sa nachádza. Opäť sa potvrdilo, že v prostredí nie je miesto ako miesto, a že túto skutočnosť možno využiť i bez priameho snaženia.

Ukázalo sa, že skupina agentov môže i vlastnými silami prekonať stagnáciu spôsobenú vnútorným „defektom“ agentov. Silu k prekonaniu stagnácie čerpá z generatívnej sily náhody.

Kapitola 10

Záver

Na záver zhrnieme myšlienky inšpirované pozorovaním modelovaných kolónií a výsledkami experimentov. Jednotlivé podkapitoly možno chápať ako návrhy na netradičné konštrukčné prvky stelesnených agentov tvoriacich kolóniu, v ktorej správaní sa majú objaviť emergentné efekty.

10.1 Interakcia

Experimenty preukázali, že i veľmi jednoduchým spôsobom môže v kolónii fungovať výrazná merateľná interakcia, bez ktorej je vznik emergentných efektov nemysliteľný. Ukázalo sa, že už len také reakcie ako je **reakcia na zrážku s iným agentom**, **reakcia na výsledok jeho činnosti** alebo **reakcia na jeho orientáciu** zabezpečia významnú interakciu.

Predpokladáme, že agenti konštruovaní v budúcnosti budú oveľa viac využívať tieto formy interakcie, ktoré boli doteraz v centre nezáujmu.

10.2 Komunikácia na diaľku

Pri experimentoch sa potvrdilo, že agenti môžu medzi sebou neadresne komunikovať, pričom komunikujúci agenti sa nemusia nachádzať na tom istom mieste, ani nemusia komunikovať v tom istom časovom okamihu.

Zdá sa, že agenti nebudú nutne potrebovať spôsob adresnej komunikácie (vysielacku a volacie znaky), ale že vystačia s **komunikáciou cez prostredie**.

10.3 Ovpływňovanie samého seba

Ukázalo sa, že i agent, u ktorého absentuje pamäť, má možnosť vplývať na seba v budúcnosti a to aktívnou zmenou svojho prostredia, na ktorú v budúcnosti reaguje.

Tento prístup k činnosti agenta sa dá vystihnúť známym porekadlom „sám sebe katom“.

10.4 Potenciálna heterogénnosť prostredia

Keď rozsypeme na zem zopár kamienkov, máme dojem, že prostredie ktoré sme vytvorili je homogénne. Ono aj je homogénne, ale len pre nás. Pre agenta s výrazne menšími rozmermi je heterogénne, pričom sú v ňom miesta potenciálne podporujúce niektoré aktivity agentov (v experimentoch to boli smetiská vyzývajúce k odhadzovaniu odpadkov a námestia vyzývajúce k zastaveniu a zhromažďovaniu).

10.5 Prostredie ako zdroj podnetov

Experimenty ukázali, že prostredie poskytuje agentom podnety nevyhnutné pre ich žiaduce správanie. Pritom spravidla nejde o podnety, ktoré by nejako súviseli s povahou riešenej úlohy.

Napríklad v experimente „Dezaktivácia územia“ bez reakcie na stretnutie s iným agentom agenti nedokázali dezaktivovať zamorené územia kompetentne.

Z toho vyplýva, že agenti musia **reagovať i na podnety nesúvisiace s povahou úlohy** a zamiešať ich vplyv na jej riešenie.

Na základe tohto prístupu možno potom rozlišovať situácie, v ktorých sú vhodné skôr deliberatívne roboty ako nedeliberatívne. Ide o prípady uniformných prostredí, ktoré nie sú zdrojom dostatočných podnetov pre funkčnú prácu reaktívnych agentov.

Keď si mravce založia mravenisko v blízkosti nejakého ľudského výtvoru, ktorý sa od výtvorov prírody spravidla líši tým, že obsahuje pre nich veľmi málo podnetov (napríklad blízko vašej terasy v záhrade), môžeme sa postaviť na hranicu medzi výtvormi ľudí a prírody a pozorovať správanie mravcov. Voľným okom budeme vidieť, že na terase mravce oveľa menej často menia smer pohybu ako na opačnej strane, dostanú sa za kratší čas oveľa ďalej. To samozrejme nie je až také neočakávané, ale práve v dôsledku tejto činnosti mravce nikdy nezaložia mravenisko v strede terasy (i keď pri múre domu, kde sa ich pohyb opäť spomaľuje, ho často stavajú).

10.6 Náhoda

Detekovať podnety z reálneho prostredia je síce možné, ale aj veľmi ťažké, lebo to zvyšuje nároky na receptory (senzory) stelesnených agentov, čo zvyšuje ich konštrukčnú náročnosť, ich nároky na zdroje energie a samozrejme ich cenu. Experimenty však

ukázali, že mnohokrát môžu byť podnety z reálneho prostredia dobre nahradené generovaním náhodných podnetov. Náhoda v podobe „šumových“ receptorov je jedným z najdôležitejších prvkov architektúry agentov v tejto práci. Ukázalo sa, že náhoda dokáže nielen čiastočne zastupovať reálne podnety, ale i vnútornú pamäť, či stav.

Uvažujme napríklad mechanickú ruku. Vieme, že ju dokážeme skonštruovať tak, aby jej pohyb funkčne zodpovedal pohybu skutočnej ruky. Pri pohľade na pohyb mechanickej ruky však cítime, že nám na ňom niečo chýba: nie je ladný ako pohyb našej ruky. Tento fakt je podľa nás spôsobený tým, že všetky súčiastky ruky sa snažia pohybovať v smere pohybu ruky. Keby sa to snažila iba väčšina súčiastok a ostatné by sa snažili hýbať inými smermi, ruka by sa pohybovala približne po tej istej trajektórii, ale pohyb by bol výslednicou protikladných pohybov. Takýto pohyb by možno mal to, čo nám chýba pri pohľade na pohyb terajších mechanických rúk.

Predpokladáme, že i keby agenti z hľadiska ich konštrukcie mali pamäť i množstvo senzorov, náhoda bude vítaným prvkom ich konštrukcie. **Náhoda bude esenciou, ktorá dá agentom správny nádyh.**

Keď pozorujeme mravčeka, ktorý sa dostal do slepej cesty medzi strmými kamienkami na ktoré nedokáže vyliezť, vidíme, že podchvíľou opakuje ten istý pohyb, ktorý ho má navrátiť späť. Ten však k úspechu nevedie a nám sa zdá, že mravček ostatne medzi kamienkami navždy. A zrazu je von. Ozvalo sa v ňom niečo, čo zmenilo jeho správanie. Presne tento efekt sa dá dosiahnuť náhodou (pozri behaviorálny modul **Moving**).

10.7 Defekty

Pod defektom rozumieme vlastnosť agentov, ktorá je v rozpore s ich naprogramovaním. Napríklad v experimente „Stretnutie na námestí“ bol na efektor **Move** vysielaný iba signál **go** a nikdy nie **stop**. Napriek tomu však po určitom čase väčšina agentov zastala a zvyšok sa pohyboval „do kruhu“.

Na defekt sa možno dívať ako na emergentný efekt. Experimenty ukázali, že zaujímavé správanie agentov sa dá dosiahnuť pôsobením takéhoto defektu v naprogramovaní agentov. Vplyv defektov má lokálne pominuteľný charakter a jeho vplyv je trvalý iba vtedy, ak zasiahne naraz podstatnú časť kolónie. Veľmi zaujímavým faktom je, že časť kolónie ktorá je postihnutá vplyvom defektu, dokáže odstrániť tento vplyv vlastnými silami. Kolónii teda môžeme prisúdiť schopnosť **zotaviť za z havárie**.

10.8 Kvantitatívne ukazovatele emergentných efektov

Z experimentov vyplýva, že intuitívne zistené emergentné efekty sú sprevádzané merateľnou zmenou kvantitatívnych ukazovateľov a naopak, túto zmenu možno vysvetliť prítomnosťou emergentného efektu. V tejto práci sme sa pokúsili vytvoriť aparát na opis tohto vzťahu. Uvedomujeme si, že tento aparát možno v mnohom vylepšiť.

Vývoj kolónie nemusíme porovnávať len s $1/2$ -kolóniou ale i s kolóniami iných veľkostí, čo by mohlo viesť k limitným záverom.

10.9 Problém rozmerov

V našich experimentoch sme sa obmedzili na objekty rovnakej veľkosti. Tým bolo zaručené, že úloha formulovaná podľa objektov prostredia bola dobre sformulovaná i pre agentov, ktorí ju mali riešiť. Pri rozdielnych rozmeroch problému a jeho riešiteľov by však dochádzalo k problémom. K problémom by dochádzalo hlavne vtedy, kedy by sa problém týkal viacerých objektov rôznych rozmerov.

Tento problém je v praxi aktuálny a jedno z možných riešení vidíme v kolóniách zložených z agentov s hierarchickou štruktúrou, keď agent na vyššej úrovni je kolóniou agentov na úrovni nižšej.

10.10 Záver záveru

Týmto sme opísali celé polienko, ktoré teraz prikladáme do ohňa umelej inteligencie a príbuzných vied.

Literatúra

[Baník 94]

Baník I.: *Two Formal Views on Colony Architecture*. Diplomová práca, Katedra umelej inteligencie, Matematicko-fyzikálna fakulta, Univerzita Komenského, Bratislava 1994.

[Brooks 86]

Brooks R. A.: *A Robust Layered Control System for a Mobile Robot*. IEEE Journal of Robotics and Automation, RA-2, (1986), 14–23.

[Brooks 89]

Brooks R. A.: *A Robots that Walks: Emergent Behaviors from a Carefully Evolved Network*. Neural Computation 1:2, Summer, (1989).

[Brooks 90]

Brooks R. A.: *The Behavior Language; User's Guide* (A. I. Memo 1227), MIT AI Lab, Cambridge, Mass., 1990.

[Doran 92]

Doran J.: *Distributed AI and its Applications* In: Advanced Topics in Artificial Intelligence (V. Mařík, O. Štěpánková, R. Trappl, eds.) Springer-Verlag, Berlin, 1992.

[Ferguson 92]

Ferguson I. A.: *TouringMachines: an Architecture for Dynamic, Rational, Mobil Agents*. (TR 273), Computer Laboratory, Cambridge University, Cambridge, 1992.

[Forest 91]

Forest S.: *Emergent computation: self-organizing, collective, and cooperative phenomena in natural and artificial computing networks*. Introduction to the Proceedings of the Ninth Annual CNLS Conference. In:

Emergent Computation (Forest S., ed.), The MIT Press, Cambridge, Mass., 1991, 1–11.

[Goodwin 93]

Goodwin R.: *Formalizing Properties of Agents* (Report CMU - CS - 93 - 159), Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1993.

[Kelemen 93]

Kelemen J.: *Multiagent symbol systems and behavior-based robots*. Applied Artificial Intelligence 7, (1993), 419–432.

[Kelemen - Kelemenová 92]

Kelemen J., Kelemenová A.: *A grammar-theoretic treatment of multi-agent systems*. Cybernetics and Systems 26, (1992), 621–633.

[Knight 93]

Knight K.: *Are Many Reactive Agents Better Than a Few Deliberative Ones?* In: Proc. IJCAI '93, Chambéry, 1993, 132–137.

[Maes 89]

Maes P.: *How To Do the Right Thing*. (A. I. Memo No. 1180), MIT AI Lab, Cambridge, Mass., 1989.

[Mataric 92]

Mataric M. J.: *Interaction and Intelligent Behavior*. (Thesis Proposal), MIT AI Lab, Cambridge, Mass., 1992.

[Mlichová 93]

Mlichová R.: *Niektoré experimenty so subsumpčnou architektúrou v nedeliberatívnej robotike*. Diplomová práca, Katedra umelej inteligencie, Matematicko-fyzikálna fakulta, Univerzita Komenského, Bratislava 1993.

[Parker 92a]

Parker L. E.: *Adaptive Action Selection for Cooperative Agent Teams*
In: From Animals to Animates, Proc. 2nd International Conference on
Simulation of Adaptive Behavior (Meyer J. A., Roirblar H., Wilson S.,
eds.), MIT Press, Cambridge, Mass., 1992, 442–450.

[Parker 92b]

Parker L. E.: *Proposal for Thesis Research in Partial Fulfillment of
the Requirements for the Degree of Doctor of Philosophy*. MIT AI Lab,
Cambridge, Mass., 1992.

Obsah

1	Úvod	5
1.1	Agent	6
1.2	Kolónia	6
1.3	Emergentné správanie	6
1.4	Cieľ práce	7
1.5	Predchádzajúce práce	7
2	Architektúra agenta	9
2.1	Základné časti agenta	9
2.2	Subsumpčná architektúra	10
2.3	Architektúra stochastického agenta	11
2.4	Komunikácia medzi agentami	15
3	Modelovanie kolónie agentov	16
3.1	Kolónia ako súčasť sveta	16
3.2	Výber akcií agenta	17
4	Matematický model kolónie agentov	19
4.1	Vzťah agentov a prostredia v čase	19
4.2	Vzťah agenta a jeho akcií v jednom okamihu	20
5	Kritérium kompetencie	23
5.1	Špecifikácia experimentu	23
5.2	Kritérium riešenia úlohy	24
5.3	Kritérium kompetentného riešenia úlohy	24
5.4	Ohraničenia spôsobené experimentálnym prístupom	26
6	Charakterizácia experimentov	27
6.1	Priestor	27
6.2	Smery a pohyb	27
6.3	Koprodukčná funkcia	28
6.4	Druhy objektov	29
6.5	Šumy	29
6.6	Konštanty	30
6.7	Receptory	30
6.8	Koprodukty, efekty a akcie	31

6.9	Receptory indikujúce stav na efektoroch	32
6.10	Behaviorálne moduly	32
7	Experiment: Dezaktivácia územia	34
7.1	Prostredie	34
7.2	Schopnosti agentov	34
7.3	Úloha	34
7.4	Riešenie: zloženie agenta a kolónie	35
7.5	Výsledky modelovania vývoja kolónie	36
7.6	Kompetentní verzus nekompetentní	49
7.7	Výsledky experimentu	57
8	Experiment: Zbieranie smetí	59
8.1	Prostredie	59
8.2	Schopnosti agentov	59
8.3	Úloha	59
8.4	Riešenie: zloženie agenta a kolónie	60
8.5	Výsledky modelovania vývoja kolónie	70
8.6	Výsledky experimentu	71
9	Experiment: Stretnutie na námestí	72
9.1	Prostredie	72
9.2	Schopnosti agentov	72
9.3	Úloha	72
9.4	Riešenie: zloženie agenta a kolónie	73
9.5	Výsledky modelovania vývoja kolónie	75
9.6	Výsledky experimentu	87
10	Záver	88
10.1	Interakcia	88
10.2	Komunikácia na diaľku	88
10.3	Ovplyvňovanie samého seba	88
10.4	Potenciálna heterogénnosť prostredia	89
10.5	Prostredie ako zdroj podnetov	89
10.6	Náhoda	89
10.7	Defekty	90
10.8	Kvantitatívne ukazovatele emergentných efektov	90
10.9	Problém rozmerov	91
10.10	Záver záveru	91
	Literatúra	92