

# Case study z predmetu „Evolučné algoritmy“

Peter Lalík, 13.1. 2000

## Zadanie

**Téma 16. Zostrojte model celulárneho automatu** so 4 stavmi (na kvalite zobrazenia nezáleží, môže byť rozlíšené rôznofarebnými štvorčekmi):

1. Prázdna bunka
2. Mravec
3. Ihličie
4. Mravec s ihličím

*Pravidlá:* Na počiatku dvojrozmerná mriežka (na okrajoch spojená do toroidu, aby neboli problémy s ohraničením) obsahuje náhodne rozmiestené ihličie a mravce. Mravec sa pohybuje vždy náhodným smerom, no tak, aby dva mravce neboli v jednej bunke. Keď narazí na ihličie, zoberie ho. Keď mravec s ihličím narazí na iné ihličie, upustí svoje ihličie. (Môže tak aj z "kôpky" odoberať.)

Pozorujte, ako sa takýmto distribuovaným procesom budú vytvárať "kôpky", až nakoniec vznikne jedna veľká kopa. Zmerajte, ako dlho to priemerne trvá, kým je väčšina ihličia na jednej kôpke.

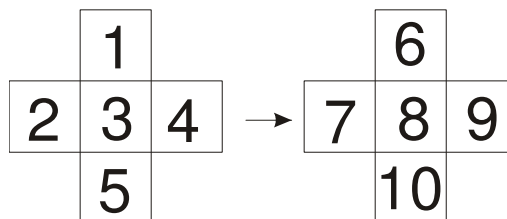
## Riešenie

Požadovaný celulárny automat  $C$  najprv definujem ako formálnu štruktúru – päťicu  $C=(N,T,P)$ , kde:

$N$  je množina neterminálnych symbolov  $\{mravec=M, mravec\ s\ ihličím=MI\}$ ,

$T$  je množina terminálov  $\{prázdna\ bunka=0, ihličie=I\}$  a

$P$  je množina pravidiel – podmnožina  $(N+T)^{10}$ , (kde  $+$  je zjednotenie), pričom prvá päťica pravidiel má význam pôvodnej a druhá nasledujúcej oblasti stavu:



(t.j., stred pôvodnej oblasti je 3 prvok, 1. je na sever, 2. na západ, 4. na východ a 5. na juh, obdobne pre prvky 6.-10., relácia „na sever“ atď je definovaná tak, aby dodržiavala spojenie na krajoch do toroidu)

„**Mriežka**“ zodpovedá dvojrozmernej matici  $M_{l..n,l..m}$ ,  $M_{i,j}$  je z  $(N+T)$ .

„**Všeobecný celulárny krok**“ je relácia medzi maticami  $M^t$  a  $M^{t+1}$ , kde  $M^{t+1}$  vznikne z  $M^t$  aplikáciou pravidiel na všetky neterminály.

„**Aplikácia pravidla  $p$  z  $P$  na neterminál**“ je prepísanie oblasti, ktorá zodpovedá  $p_{1..5}$  na oblasť zodpovedajúcu  $p_{6..10}$ . V prípade, že na neterminál  $M_{i,j}^t$  nie je aplikovateľné žiadne pravidlo z  $P$ , platí  $M_{i,j}^{t+1} = M_{i,j}^t$ .

„**Celulárny krok**“ je taký všeobecný celulárny krok, v ktorom „nemiznú mravce“, t.j. počet symbolov z  $\{M,MI\}$  je rovnaký v  $M^t$  aj  $M^{t+1}$ . (Mravce môžu „zmiznúť“ aplikáciou pravidiel, keď sa viac mravcov pohne na isté (voľné) miesto.)

Zadanému problému zodpovedajú tieto pravidlá ( $x, y, z$  sú ľubovoľné symboly):

Pohyb mravca:

$(0, x, M, y, z,$	$M, x, 0, y, z)$	(na sever)
$(x, 0, M, y, z,$	$x, M, 0, y, z)$	(na západ)
$(x, y, M, 0, z,$	$x, y, 0, M, z)$	(na východ)
$(x, y, M, z, 0,$	$x, y, 0, z, M)$	(na juh)

$(0, x, MI, y, z,$	$MI, x, 0, y, z)$	(na sever)
$(x, 0, MI, y, z,$	$x, MI, 0, y, z)$	(na západ)
$(x, y, MI, 0, z,$	$x, y, 0, MI, z)$	(na východ)
$(x, y, MI, z, 0,$	$x, y, 0, z, MI)$	(na juh)

Zobratie ihličia:

$(I, x, M, y, z,$	$MI, x, 0, y, z)$	(na sever)
$(x, I, M, y, z,$	$x, MI, 0, y, z)$	(na západ)
$(x, y, MI, I, z,$	$x, y, 0, MI, z)$	(na východ)
$(x, y, M, z, I,$	$x, y, 0, z, MI)$	(na juh)

Výmena ihličia:

$(I, x, MI, y, z,$	$MI, x, I, y, z)$	(na sever)
$(x, I, MI, y, z,$	$x, MI, I, y, z)$	(na západ)
$(x, y, MI, I, z,$	$x, y, I, MI, z)$	(na východ)
$(x, y, MI, z, I,$	$x, y, I, z, MI)$	(na juh)

Z formálneho hľadiska je úlohou zostrojovať postupnosti  $(M^0, M^1, \dots)$ , kde  $M^0$  je náhodná mriežka a  $M^i$  s  $M^{i+1}$  sú v relácii „celulárny krok“ a na týchto postupnostiach pozorovať emergentné javy.

Z praktického hľadiska je lepšie zohľadniť špecifiká úlohy a v implementácii sa vyhnúť sa zbytočne všeobecnej a menej efektívnej aplikácii celulárneho automatu. Konkrétne, napríklad proces aplikácie pravidla je v programe riešený s ohľadom na typ pravidiel (pohyb) s predchádzaním kolízií mravcov.

Oproti formálnemu popisu má implementácia ešte jednu odlišnosť: umožňuje pohyb mravcom aj na miesto, ktoré sa v rovnakom kroku uvoľnilo (riziko „miznutia mravcov“). Toto by nebolo možné pri aplikácii pravidiel.

Rutina simulujúca (s vyššie uvedenou modifikáciou) pohyb mravcov je teda implementovaná takto:

```
procedure MoveAnts;
var
  NewMatrix : TMatrix;
  x, y, n, a : integer;
  ed : array[1..8] of integer;
begin
  {skopirujeme ihlicie;}
  for x:=0 to MatrixWidth-1 do
  for y:=0 to MatrixHeight-1 do
  if Matrix[x,y] = Needle then
    NewMatrix[x,y]:=Matrix[x,y] else
    NewMatrix[x,y]:=Empty;
  {posunieme mravce;}
  for x:=0 to MatrixWidth-1 do
  for y:=0 to MatrixHeight-1 do
  if Matrix[x,y] in [Ant,AntNeedle] then
  begin
    n:=0;
    {zistime mozne smery;}
    for a:=1 to 4 do
      if NewMatrix[DirX(x,a),DirY(y,a)] in [Empty,Needle] then
      begin
        {pridame povoleny smer;}
        inc(n);
        ed[n]:=a;
      end;
    {ak sa moze pohnut, pohne sa, inak ostava;}
    if n>0 then
```

```

begin
  a:=ed[random(n)+1];
  {ak narazi na ihlicie a uz nesie, pusti a chyti nove;}
  if (Matrix[x,y]=AntNeedle) and
    (NewMatrix[DirX(x,a),DirY(y,a)] = Needle) then
    NewMatrix[x,y] := Needle;
  if NewMatrix[DirX(x,a),DirY(y,a)] = Needle then
    NewMatrix[DirX(x,a),DirY(y,a)] := AntNeedle else
    NewMatrix[DirX(x,a),DirY(y,a)] := Matrix[x,y];
  end else
    NewMatrix[x,y]:=Matrix[x,y];
  end;
  Matrix:=NewMatrix;
  RefreshMatrix;
end;

```

(funkcie DirX a DirY vracajú súradnice daným smerom od zadaného koordinátu)

V takto definovanej simulácii však nie je reálne očakávať emergenciu kôpok. Intuitívna úvaha: akonáhle mravec narazí na ihličie, už sa ho nezbaví (môže ho len „vymeniť za iné“), pričom má naďalej neobmedzený pohyb, aj po kôpkach ihličia. Na celú situáciu sa teda môžeme pozeriť tak, že v skutočnosti sa kvázináhodne (pomocou prechádzajúcich mravcov) posúva ihličie. Jediná stochastická závislosť je determinovaná prítomnosťou mravca (t.j. čím je ihličie bližšie k niektorému mravcovi, tým je pravdepodobnejšie, že sa v blízkej budúcnosti pohne). Okrem toho však úplnej náhodnosti pohybu ihličia nič nebráni.

Reálnym experimentom s takto definovanými pravidlami sa táto hypotéza potvrdila, keď sa ani po rádovo niekoľko sto tisíc krokoch nevyskytol jav pripomínajúci kumuláciu ihličia.

Je viacero spôsobov, ako docieľiť, aby v systéme dochádzalo k želanej emergencii. Všetky spôsoby majú spoločnú snahu prinútiť mravce zanechávať ihličie „na tých správnych miestach“, t.j. na okrajoch kôpok. Ako najjednoduchší spôsob modifikácie doterajšieho modelu tak aby sa naplnil tento cieľ sa javí zmeniť pravidlá  $P$  tak, aby mravce mohli nesené ihličie zanechať, a to vtedy, ak narazia na kraj kôpky. Vhodné je tiež, ak sa mravce nebudú môcť pohybovať po ihličí, t.j. zrušiť pravidlá „výmena ihličia“.

Celkovo, úpravy v pravidlách sú:

- zrušenie skupiny pravidiel „výmena ihličia“
- prídanie pravidiel, umožňujúcich zanechanie ihličia, ak mravec „narazil na iné ihličie“:

#### Uloženie ihličia:

$(0,x,MI,y,z, \quad M,x,0,y,z)$  (na sever)  
 $(x,0,MI,y,z, \quad x,M,0,y,z)$  (na západ)  
 $(x,y,MI,0,z, \quad x,y,0,M,z)$  (na východ)  
 $(x,y,MI,z,0, \quad x,y,0,z,M)$  (na juh)

kde v každom pravidle platí:  $x=I$  alebo  $y=I$  alebo  $z=I$ .

Implementácia týchto pravidiel vyzerá takto:

```

procedure MoveAnts;
var
  NewMatrix : TMatrix;
  x,y,n,a : integer;
  ed : array[1..8] of integer;
  NearNeedle : boolean;
begin
  {skopirujeme ihlicie;}
  for x:=0 to MatrixWidth-1 do
    for y:=0 to MatrixHeight-1 do
      if Matrix[x,y] = Needle then
        NewMatrix[x,y]:=Matrix[x,y] else
        NewMatrix[x,y]:=Empty;
  {posunieme mravce;}
  for x:=0 to MatrixWidth-1 do
    for y:=0 to MatrixHeight-1 do
      if Matrix[x,y] in [Ant,AntNeedle] then
        begin
          NearNeedle:=false;
          for a:=1 to 4 do
            if NewMatrix[DirX(x,a),DirY(y,a)] = Needle then NearNeedle:=true;
            if Matrix[x,y] = Ant then
              begin
                n:=0;
                for a:=1 to 4 do
                  if (NewMatrix[DirX(x,a),DirY(y,a)] in [Empty,Needle]) and
                    (Matrix[DirX(x,a),DirY(y,a)] in [Empty,Needle]) then
                    begin inc(n); ed[n]:=a; end;
                end else
                begin

```

```

{s ihlicim nemoze prejst na ine ihlicie}
n:=0;
for a:=1 to 4 do
  if (NewMatrix[DirX(x,a),DirY(y,a)] = Empty) and
    (Matrix[DirX(x,a),DirY(y,a)] = Empty) then
    begin inc(n); ed[n]:=a; end;
end;
{ak sa moze pohnut, pohne sa, inak ostava;}
if n>0 then
begin
a:=ed[random(n)+1];
{ak je pri ihlici a uz nesie, moze pustit a "odist bez" s pravdep. 50%;}
if Matrix[x,y]=AntNeedle then
begin
  if NearNeedle and (NewMatrix[x,y]=Empty) and (random(2)=1) then
  begin
    NewMatrix[x,y] := Needle;
    NewMatrix[DirX(x,a),DirY(y,a)] := Ant; {!predp., ze ide na Empty}
  end else
    NewMatrix[DirX(x,a),DirY(y,a)] := Matrix[x,y]; {!predp., ze ide na Empty}
  end else
    if NewMatrix[DirX(x,a),DirY(y,a)] = Needle then
      NewMatrix[DirX(x,a),DirY(y,a)] := AntNeedle else
      NewMatrix[DirX(x,a),DirY(y,a)] := Matrix[x,y];
    end else
      NewMatrix[x,y]:=Matrix[x,y];
    end;
  Matrix:=NewMatrix;
  RefreshMatrix;
end;

```

V tomto stave už systém prejavuje emergentné znaky zhromažďovania ihličia.

## Popis programu

Aplikácia (CAAnts.PAS) je naprogramovaná v Borland Pascale 7. Pracuje v dvoch režimoch – v grafickom, kde užívateľ vidí mriežku a animovaný pohyb mravcov a v textovom „dialógovom“ móde, kde môže modifikovať parametre a získavať informácie o stave systému.

Priebeh programu sa zaznamenáva do log-súboru (CAAnts.LOG).

Po spustení programu je automaticky nastavený grafický mód a spustená simulácia. Ihličie je zobrazené zelenými obdĺžnikmi, mravce bielymi, mravce s ihličím majú zelený pás.

Po stlačení klávesy Esc sa zapne textový dialógový režim. K dispozícii sú tieto príkazy:

- ENTER pokračovanie v grafickej simulácii
- X ukončenie programu
- ? nápoveda
- I informácie o stave (zapisuje sa aj do log-súboru), ako číslo generácie (kroku), počet mravcov, ihličia a pod.
- R reštart simulácie so zadanými parametrami (počet mravcov a ihličia)
- C pokračuje v simulácii bez grafického zobrazenia, je možné nastaviť priebežné vyhodnocovanie počtu kôpok v zadaných intervaloch (čo sa zaznamenáva do log-súboru)
- P grafické zobrazenie aktuálneho stavu mriežky
- M vyhodnotí počet kôpok a ich veľkosti
- L umožní užívateľovi zapísať riadok do log-súboru
- D zmení grafický mód pre simuláciu (buď pomalší 16-farebný, kde funguje screen-capture, alebo rýchlejší 256-farebný; oba v rozlíšení 640x480)

Veľkosť mriežky sa nedá nastavovať počas behu programu, je zadaná konštantami priamo v programe (MatrixWidth, MatrixHeight). Pri ich zmene je vhodné zmeniť aj veľkosť grafickej bunky (GraphicCell).

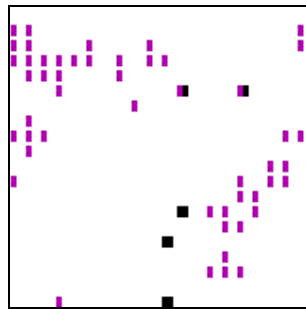
## Experimenty

Hoci systém zjavne konverguje k vytváraniu kôpok ihličia, nevzniká nakoniec jedna veľká kopa, častejší je (pokiaľ je množstvo ihličia rádovo (resp. aspoň trojnásobne) vyššie ako počet mravcov) vznik mnohých drobných kôpok. Druhé pozorovanie je, že rýchlosť utvárania a veľkosti kôpok výrazne závisia na počte mravcov, ihličia a veľkosti mriežky.

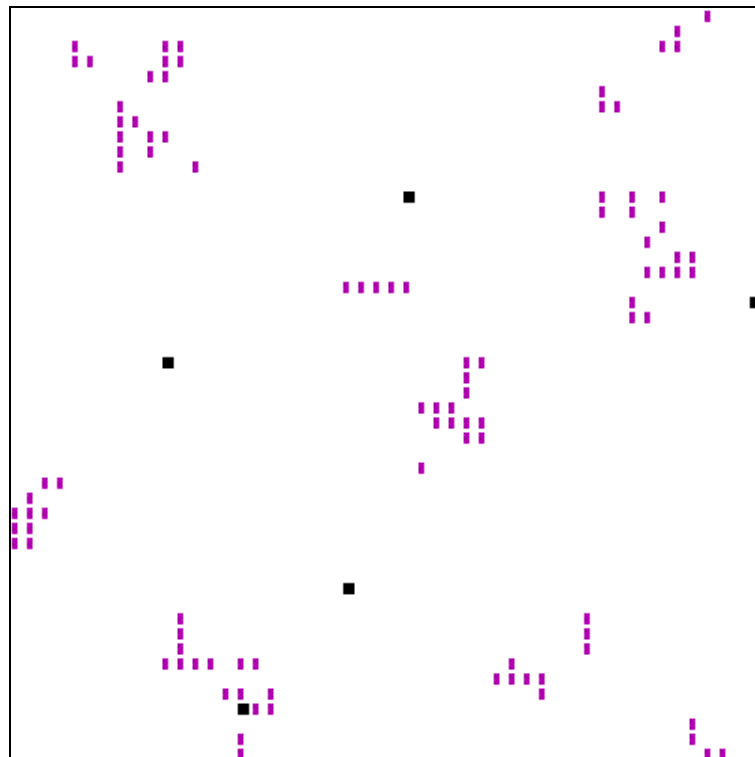
Prevedené merania:

1. 20x20, 5 mravcov, 50 ihličia
2. 50x50, 5 mravcov, 100 ihličia
3. 50x50, 15 mravcov, 100 ihličia
4. 50x50, 15 mravcov, 200 ihličia
5. 50x50, 15 mravcov, 400 ihličia
6. 50x50, 30 mravcov, 100 ihličia
7. 50x50, 30 mravcov, 200 ihličia
8. 50x50, 30 mravcov, 400 ihličia
9. 100x100, 100 mravcov, 600 ihličia
10. 100x100, 50 mravcov, 600 ihličia

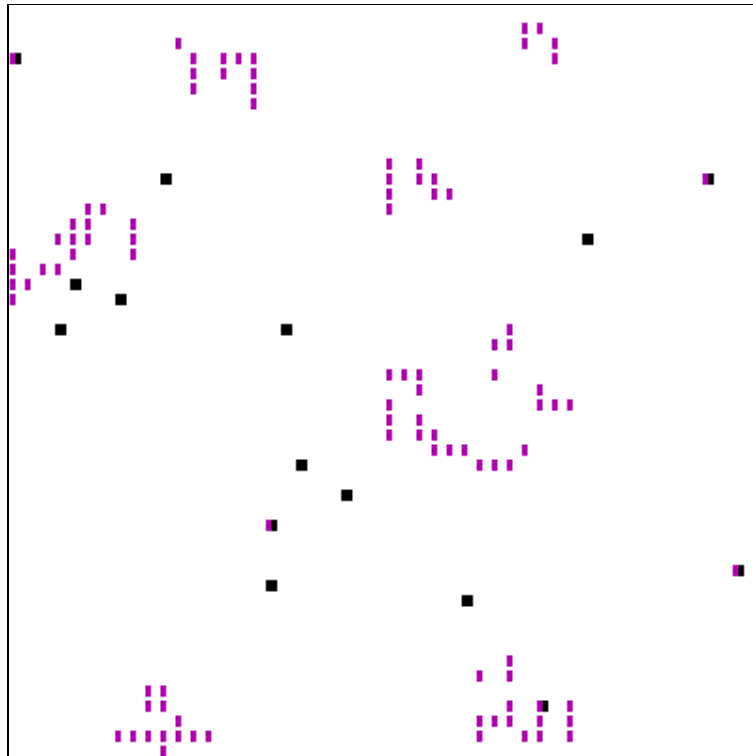
Nasledujú obrázky stavu simulácie pre jednotlivé experimenty, keď bola ukončená, grafy a tabuľky vybraných hodnôt:



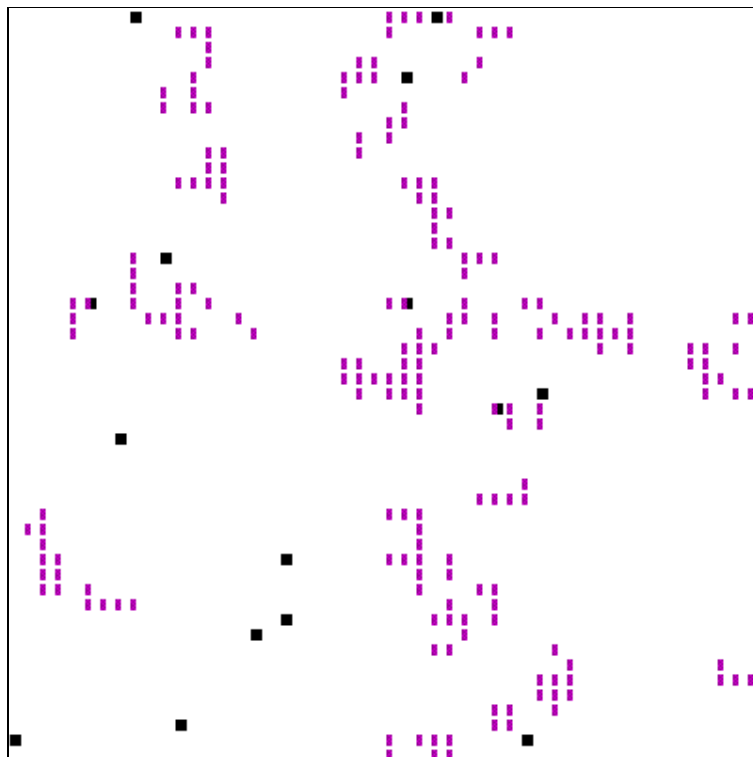
**Experiment 1**



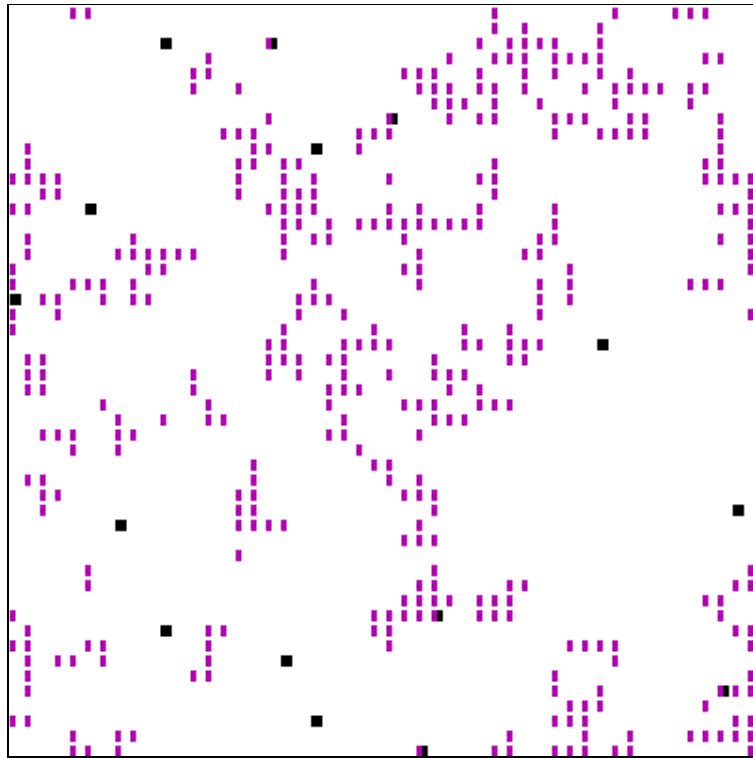
**Experiment 2**



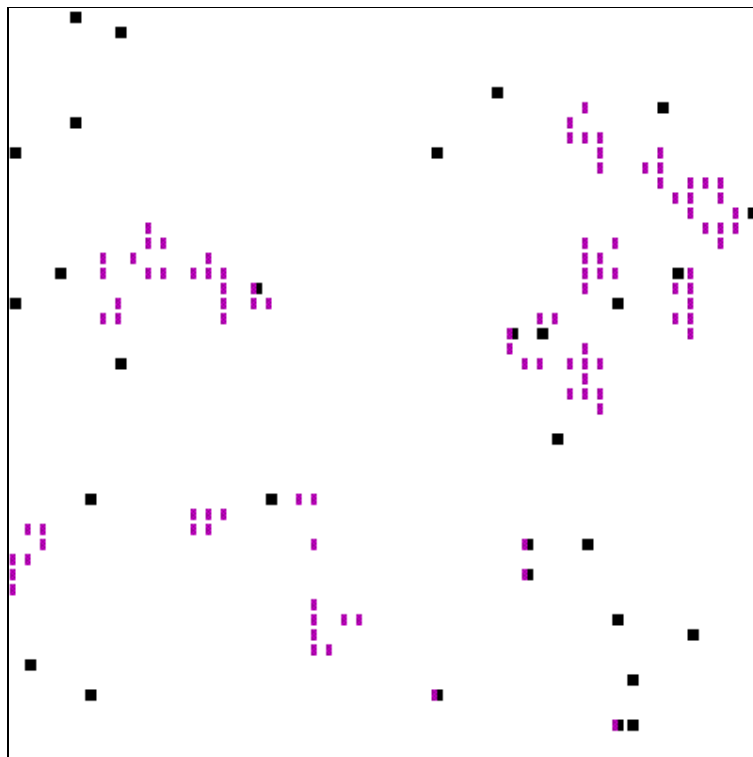
Experiment 3



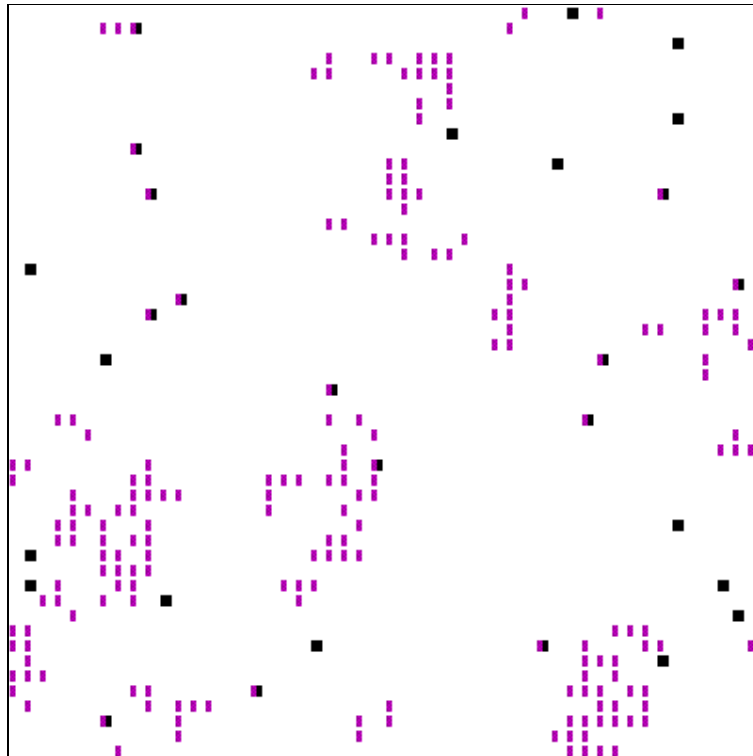
Experiment 4



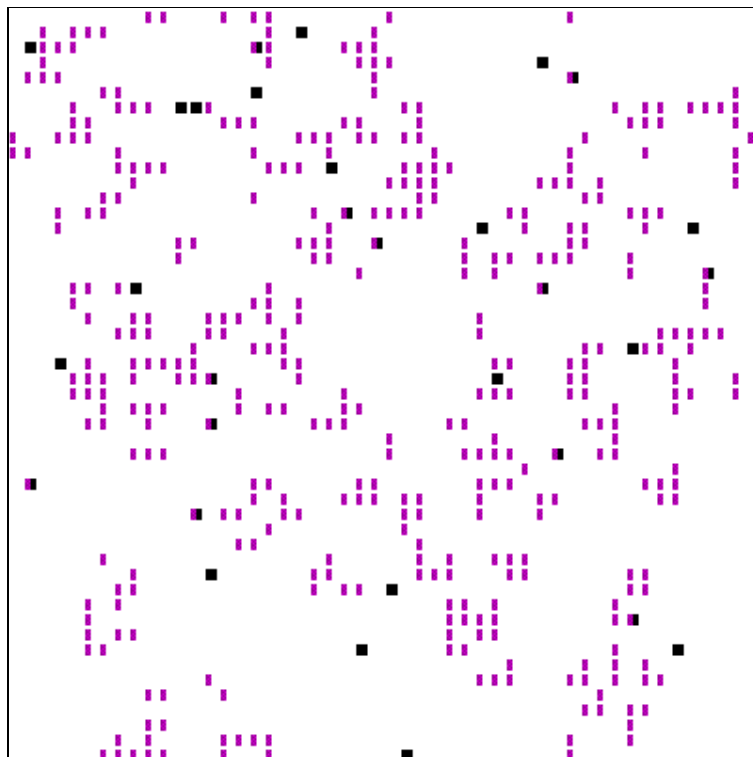
Experiment 5



Experiment 6

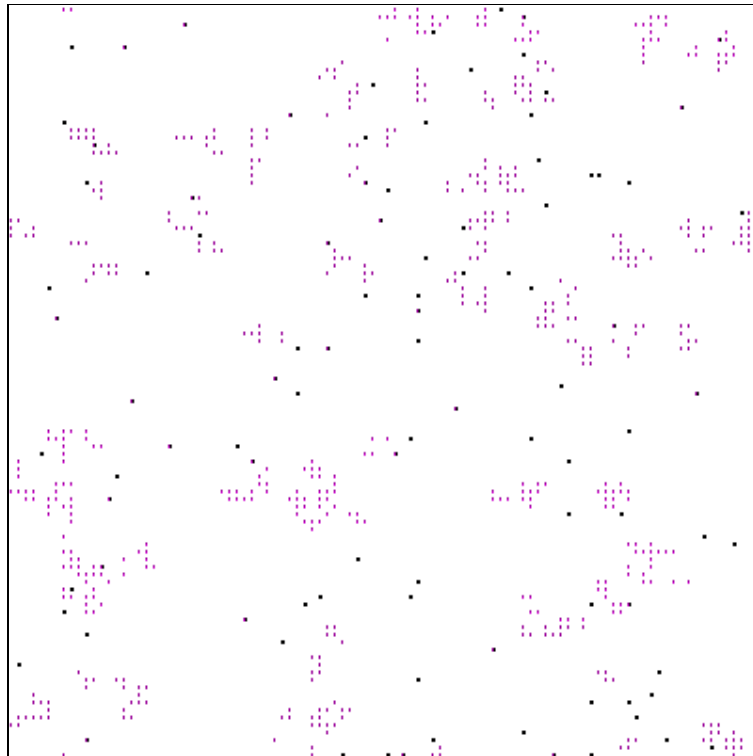


Experiment 7

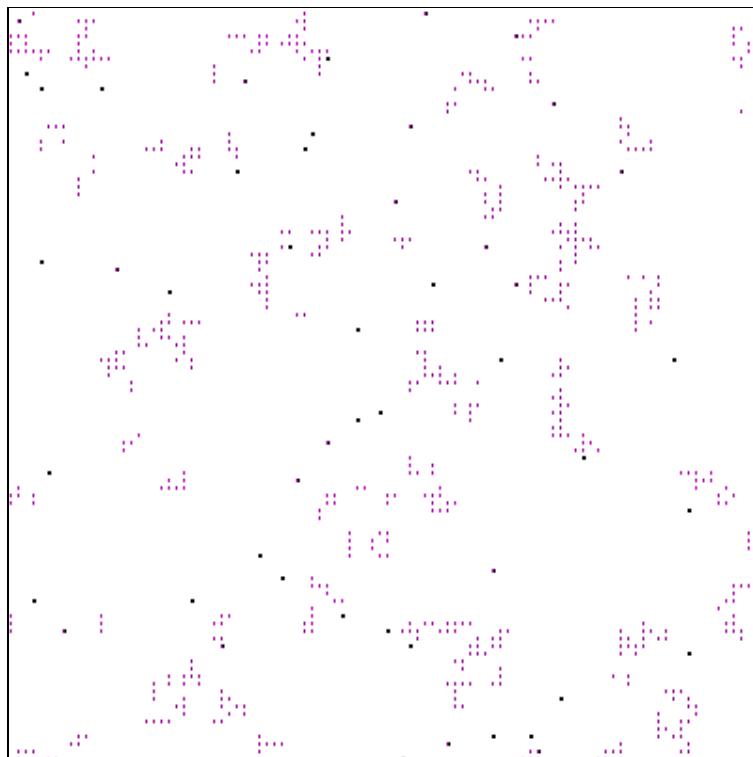


Experiment 8



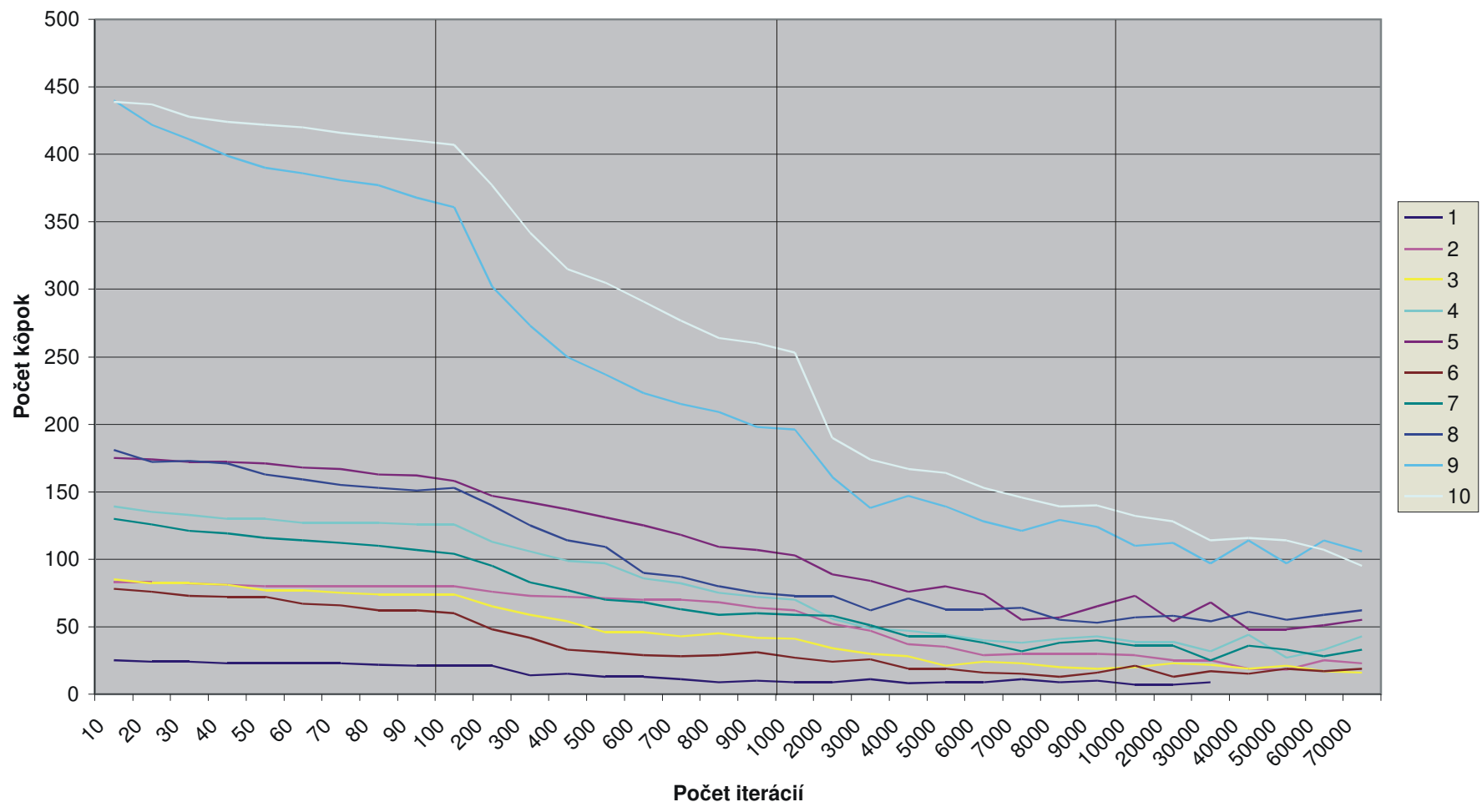


**Experiment 9**

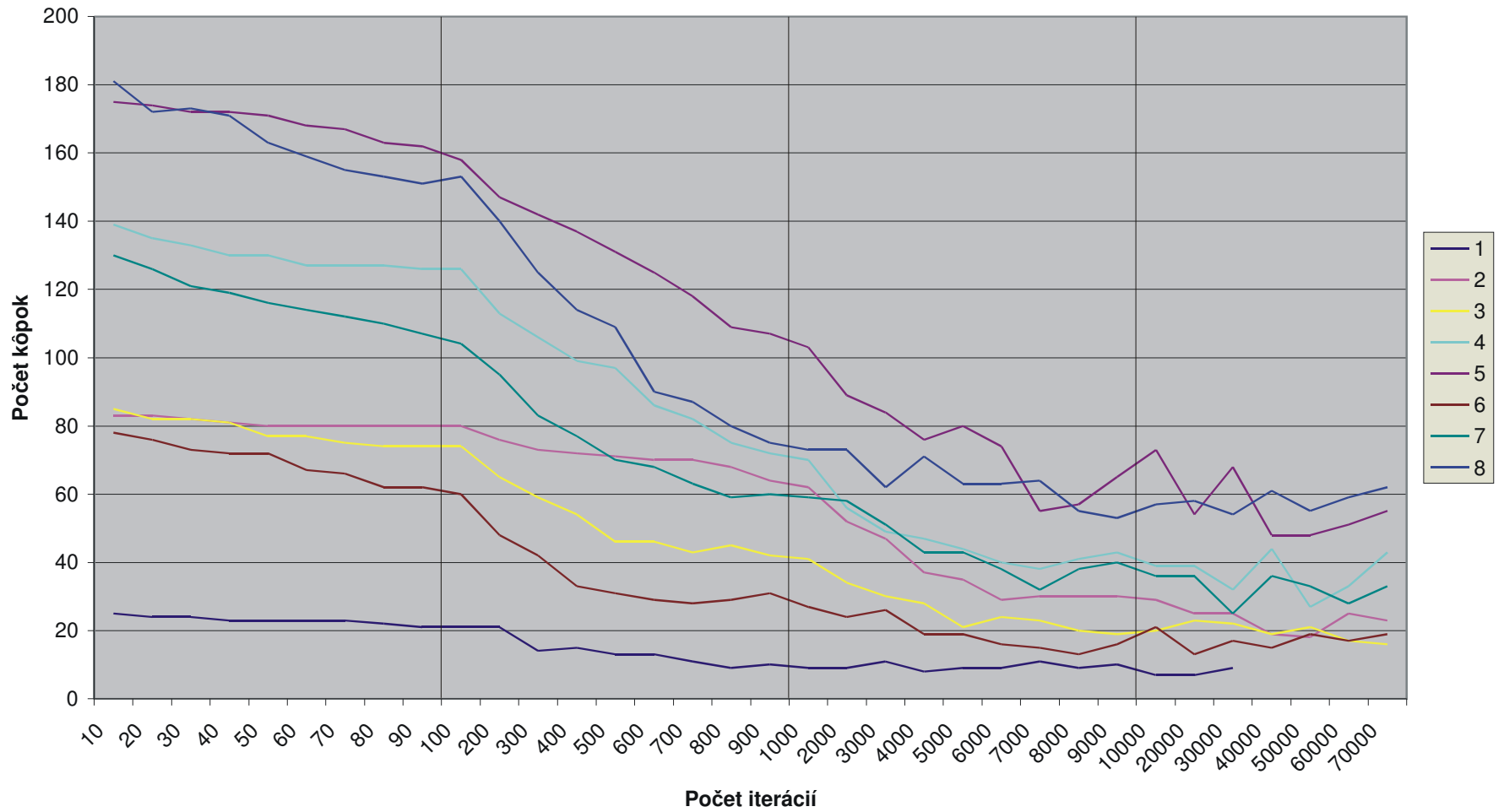


**Experiment 10**

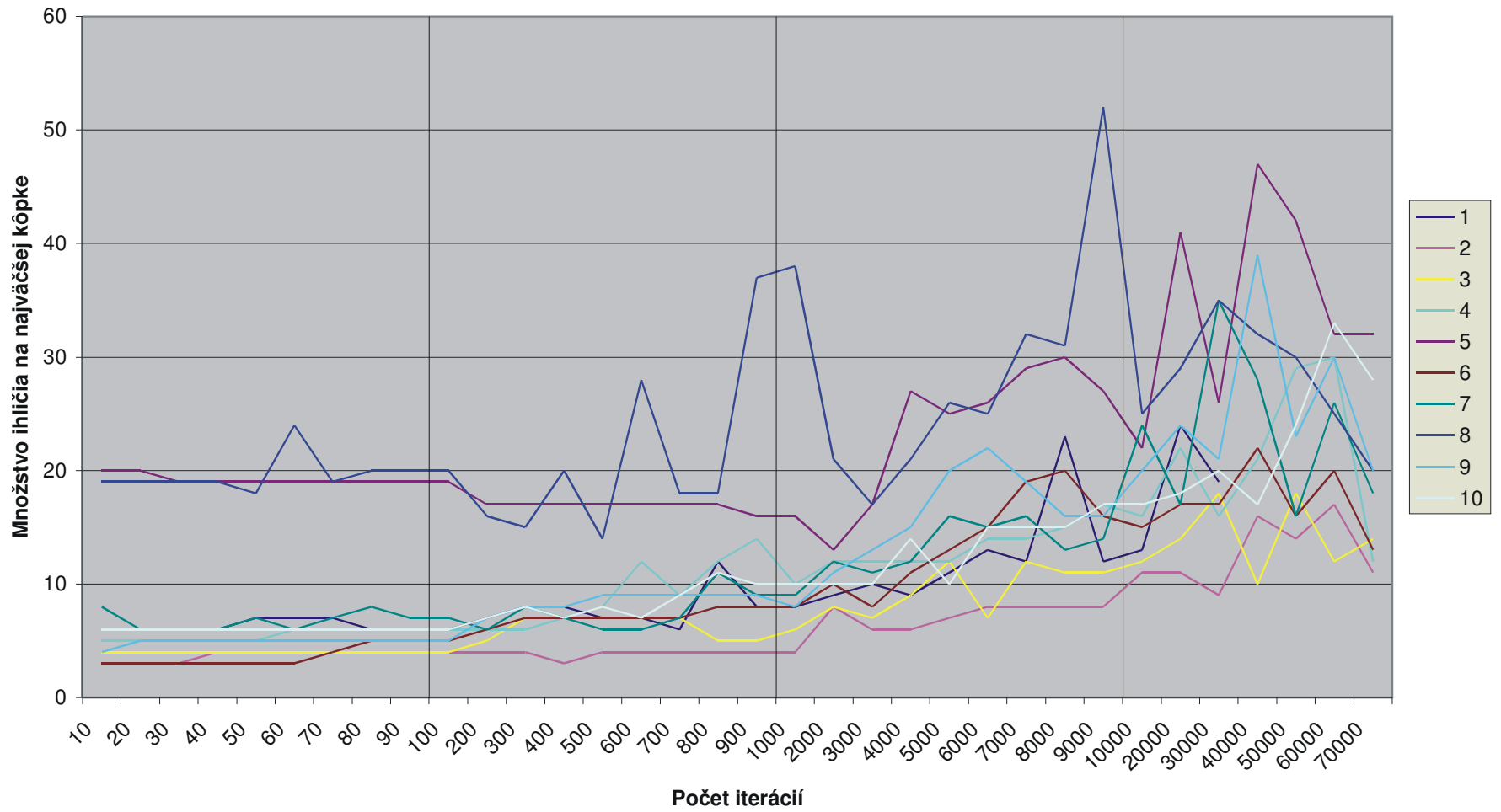
### Vývoj počtu kôpok v experimentoch



Vývoj počtu kôpok v experimentoch 1 až 7



### Vývoj veľkosti najväčšej kôpky v experimentoch



Kôpky	1	2	3	4	5	6	7	8	9	10
10	25	83	85	139	175	78	130	181	440	439
20	24	83	82	135	174	76	126	172	422	437
30	24	82	82	133	172	73	121	173	411	428
40	23	81	81	130	172	72	119	171	399	424
50	23	80	77	130	171	72	116	163	390	422
60	23	80	77	127	168	67	114	159	386	420
70	23	80	75	127	167	66	112	155	381	416
80	22	80	74	127	163	62	110	153	377	413
90	21	80	74	126	162	62	107	151	368	410
100	21	80	74	126	158	60	104	153	361	407
200	21	76	65	113	147	48	95	140	302	377
300	14	73	59	106	142	42	83	125	273	342
400	15	72	54	99	137	33	77	114	250	315
500	13	71	46	97	131	31	70	109	237	305
600	13	70	46	86	125	29	68	90	223	291
700	11	70	43	82	118	28	63	87	215	277
800	9	68	45	75	109	29	59	80	209	264
900	10	64	42	72	107	31	60	75	198	260
1000	9	62	41	70	103	27	59	73	196	253
2000	9	52	34	56	89	24	58	73	161	190
3000	11	47	30	49	84	26	51	62	138	174
4000	8	37	28	47	76	19	43	71	147	167
5000	9	35	21	44	80	19	43	63	139	164
6000	9	29	24	40	74	16	38	63	128	153
7000	11	30	23	38	55	15	32	64	121	146
8000	9	30	20	41	57	13	38	55	129	139
9000	10	30	19	43	65	16	40	53	124	140
10000	7	29	20	39	73	21	36	57	110	132
20000	7	25	23	39	54	13	36	58	112	128
30000	9	25	22	32	68	17	25	54	97	114
40000		19	19	44	48	15	36	61	114	116
50000		18	21	27	48	19	33	55	97	114
60000		25	17	33	51	17	28	59	114	107
70000		23	16	43	55	19	33	62	106	95

Max. kopa	1	2	3	4	5	6	7	8	9	10
10	6	3	4	5	20	3	8	19	4	6
20	6	3	4	5	20	3	6	19	5	6
30	6	3	4	5	19	3	6	19	5	6
40	6	4	4	5	19	3	6	19	5	6
50	7	4	4	5	19	3	7	18	5	6
60	7	4	4	6	19	3	6	24	5	6
70	7	4	4	6	19	4	7	19	5	6
80	6	4	4	6	19	5	8	20	5	6
90	6	4	4	6	19	5	7	20	5	6
100	6	4	4	6	19	5	7	20	5	6
200	7	4	5	6	17	6	6	16	7	7
300	8	4	7	6	17	7	8	15	8	8
400	8	3	7	7	17	7	7	20	8	7
500	7	4	7	8	17	7	6	14	9	8
600	7	4	7	12	17	7	6	28	9	7
700	6	4	7	9	17	7	7	18	9	9
800	12	4	5	12	17	8	11	18	9	11
900	8	4	5	14	16	8	9	37	9	10
1000	8	4	6	10	16	8	9	38	8	10
2000	9	8	8	12	13	10	12	21	11	10
3000	10	6	7	12	17	8	11	17	13	10
4000	9	6	9	12	27	11	12	21	15	14
5000	11	7	12	12	25	13	16	26	20	10
6000	13	8	7	14	26	15	15	25	22	15
7000	12	8	12	14	29	19	16	32	19	15
8000	23	8	11	15	30	20	13	31	16	15
9000	12	8	11	17	27	16	14	52	16	17
10000	13	11	12	16	22	15	24	25	20	17
20000	24	11	14	22	41	17	17	29	24	18
30000	19	9	18	16	26	17	35	35	21	20
40000		16	10	21	47	22	28	32	39	17
50000		14	18	29	42	16	16	30	23	24
60000		17	12	30	32	20	26	25	30	33
70000		11	14	12	32	13	18	20	20	28

Poznámky k prezentovaným výsledkom:

- Obrázky mriežky sú invertované, t.j. mravce sú čierne, ihličie je fialové a mravec s ihličím je fialovočierny.
- Parametre na x-ovej súradnici grafov stúpajú lineárne po častiach (10 až 100, 100 až 1000, 1000 až 10000 a nad 10000).
- V prvom experimente, na rozdiel od ostatných, neprekročil počet iterácií 40000.
- Grafy, ako aj k nim zodpovedajúca tabuľka, zodpovedajú len vybraným údajom, všetky merané údaje sú v súbore EXPRMNTS.LOG
- Za kôpky sú považované zoskupenia ihličia spojité z hľadiska 8-okolia bunky, ihličie nesené mravcami nie je brané do úvahy.

## Interpretácia výsledkov

Pozorovaný systém prejavuje v ďalšom texte popísané zákonitosti. Ich popis je skôr vágny, vyjadruje skôr zásadné trendy vývoja systému bez formalizácie a presného číselného spracovania. Pozorovania sa týkajú javov po zmene uvedených parametrov v rámci jednej mriežky (t.j. jej rozmery sa pri danej zmene počtu ihličia alebo mravcov nemenia).

Pre konštantnú veľkosť mriežky výsledný počet kôpok závisí (pri zachovaní pomeru ihličie/mravec) od hustoty (počtu) ihličia. V rozsahu experimentov (hustota 4%, 8% a 16%) sa tento vzťah javil približne lineárny, aj keď správanie mimo tohto rozsahu bude pravdepodobne iné. Ako je naznačené a bude naznačené aj nižšie, počet kôpok závisí aj od pomeru počtu mravcov a kôpok, pričom k výraznej redukcii dochádza až s priblížením sa pomeru k 1 (vo vyššie prezentovaných experimentoch bolo ihličia vždy rádovo viac).

Počet kôpok kvázi-konverguje k výslednému počtu, pričom rýchlosť kvázi-konvergenencie závisí od počtu mravcov (čím ich je viac, tým rýchlejšie). „Kvázi“ znamená, že okolo „cieľového“ počtu reálny počet osciluje. Konvergenca je teda nestabilná a miera jej nestability tiež závisí od počtu mravcov (čím ich je menej, tým je systém stabilnejší).

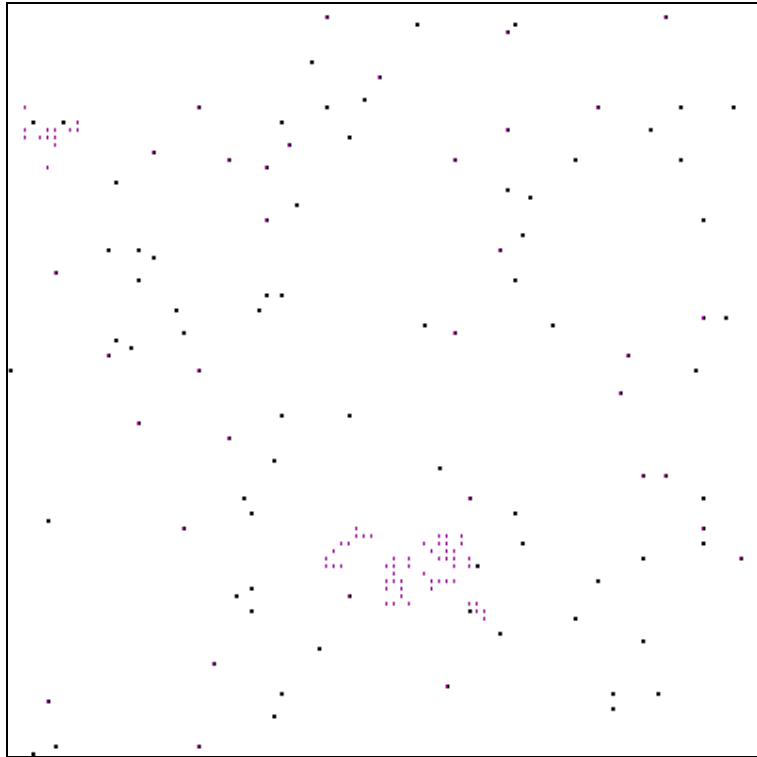
Veľkosť maximálnej kôpky je údajom ešte menej stabilným, aj keď celkovo je badať stúpajúci trend, ktorý sa časom mení na výraznú osciláciu okolo jednej priemernej hodnoty. Podobne ako pri počte kôpok, so zvýšením počtu mravcov sa miera oscilácie zvyšuje a konečná („limitná“) priemerná hodnota je úmerne závislá od hustoty ihličia.

Tieto úvahy sa týkajú situácie, keď je ihličia rádovo viac ako mravcov. V prípade, že je počet ihličia a mravcov porovnateľný, tendencia je postupné vytváranie jednej-dvoch oblastí (väčšinou nie striktné spojitéch, hoci vizuálne jasne vyčleniteľných). Tento jav je spôsobený výraznou počiatočnou redukciou „noseného ihličia“, čo vedie k vytvoreniu len niekoľkých izolovaných miest, kde sa môže ihličie kumulovať. Proces na princípe „veľká kopa pýta viac“ je tu oproti prípadu prebytku ihličia zvýraznený relatívnym nedostatkom ihličia na ploche, ktorý umožňuje mravcom putovať priamo ku „kumulatívnym centráram“. Pre ilustráciu vid' obrázky ďalších experimentov:

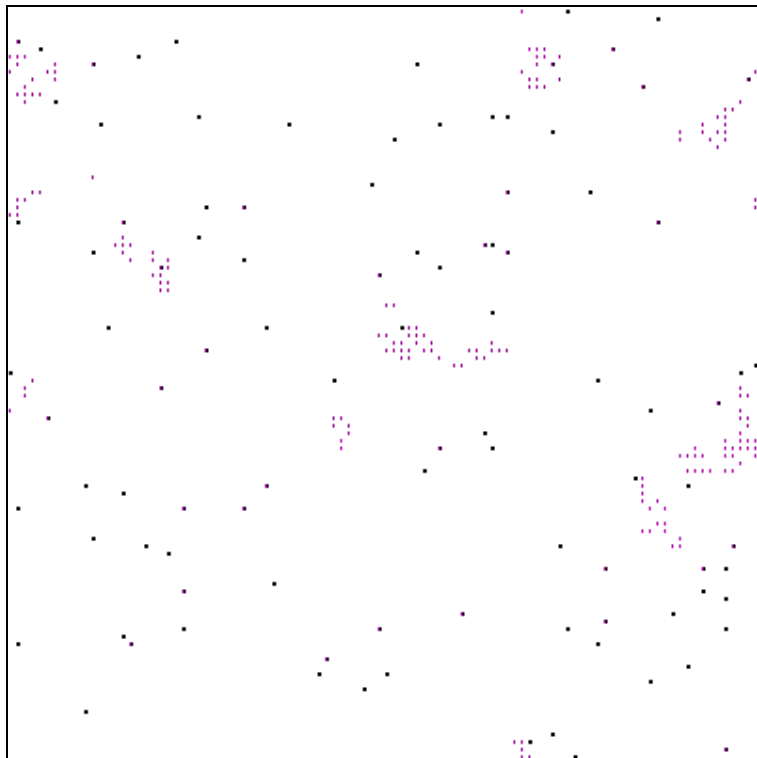
11. 100x100, 100 mravcov, 100 ihličia

12. 100x100, 100 mravcov, 200 ihličia

(obrázky sú v stave po rádovo niekoľko sto tisíc iteráciách)



**Experiment 11**



**Experiment 12**

Vo všeobecnosti možno povedať, že precíznejšia teória, z ktorej by vyplývali exaktnejšie formulované vzťahy počtu a veľkosti kôpok od veľkosti mriežky, množstva ihličia a mravcov a počtu iterácií si pravdepodobne vyžaduje špeciálne definované objekty a miery nad nimi, s pravdepodobnostnými transformáciami na nich, čo do komplexity prekračuje rámec tejto práce.

## Súbory k práci

Spakovaný súbor CS1.ZIP obsahuje:

CS1.DOC	táto správa vo formáte pre MS Word 97
CS1.PS	táto správa v PostScripte
CAANTS.PAS	program
CAANTS.EXE	skompilovaný program
CAANTS.LOG	log-súbor so záznamom priebehu experimentov
EGAVGA.BGI	grafický ovládač
EXPRMNTS.LOG	log-súbor s prezentovanými experimentami
EXPRMNTS.XLS	spracovanie výsledkov (súbor MS Excel-u)
LOG2TAB.PAS	nástroj na extrakciu dát z log-súboru
SVGA256.BGI	grafický ovládač