

Fig. 1: Patterns (species)

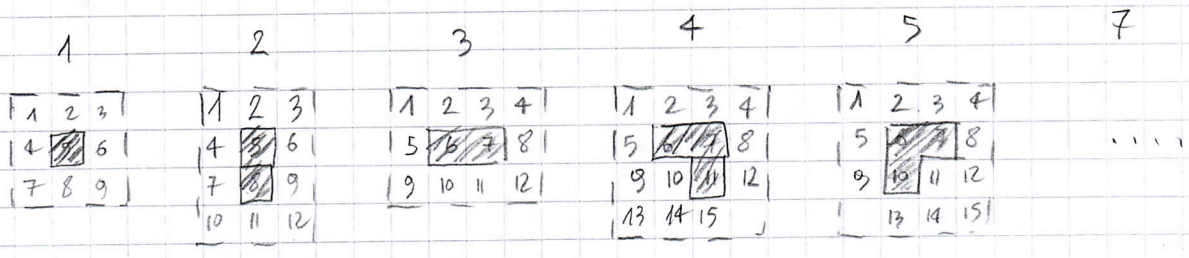
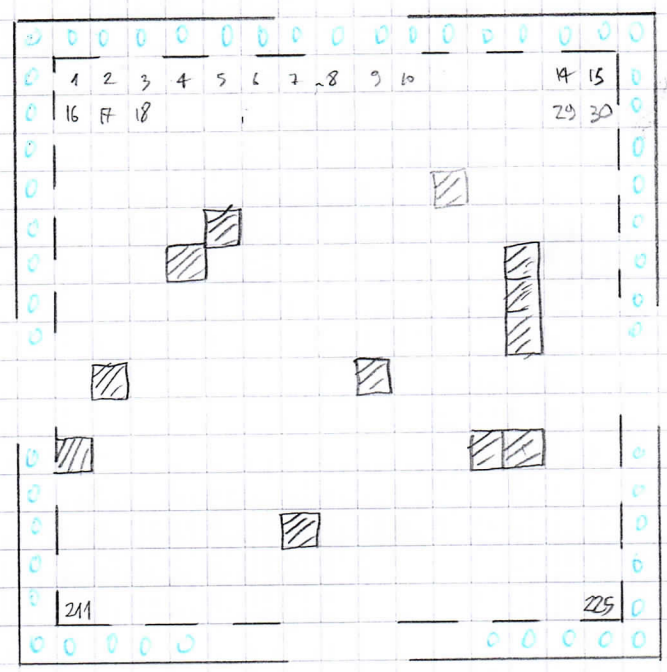
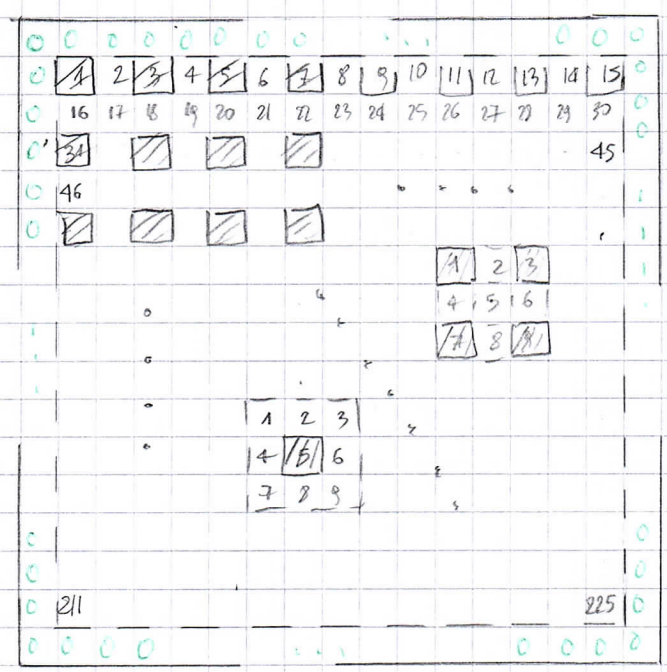


Fig. 2: CA[M+1,N+1]

initial state (clear=0)



final state: option A (only species 1)



final state: option

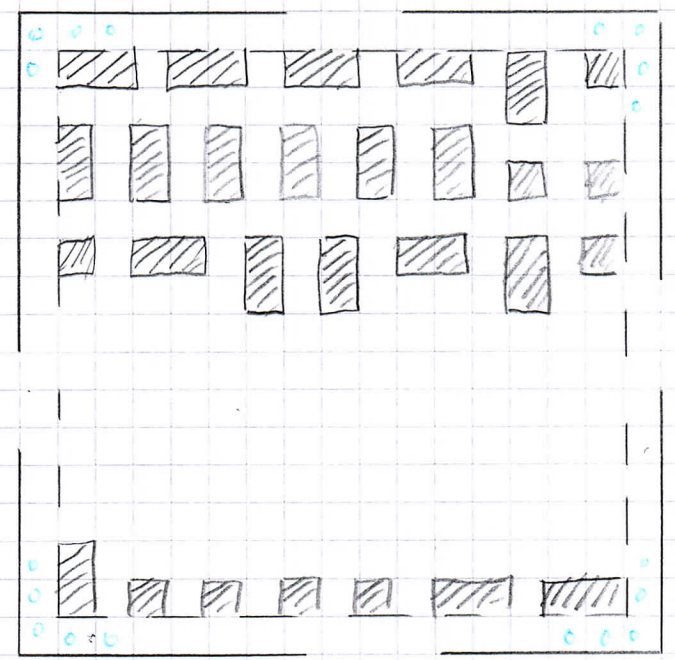


Fig. 3

2/6

- each agent-player plays a game with each of his 8 neighbours, and receives in each game a payoff

$$u_{ij}(s_{ij}, s_{ikjk})$$

according to the payoff function

- he cumulates his payoff from 8 games:

$$\sum_{k=1}^8 u_{ij}(s_{ij}, s_{ikjk})$$

and his average cumulated payoff is equal to

$$\sum_{k=1}^8 u_{ij}(s_{ij}, s_{ikjk}) / 8$$

- each player wants to maximize his payoff, but we will be interested in maximization of the average total payoff $\bar{u}()$ of the whole set of players

$$\bar{u}(s_{11}, s_{12}, \dots, s_{m,n}) = \frac{1}{m \cdot n} \sum_{j=1}^m \sum_{i=1}^n \sum_{k=1}^8 u_{ij}(s_{ij}, s_{ikjk}) / 8$$

Fig. 4

Payoff function of (i,j) -th player when we have to do with ONLY species 1

opponent player
(player: 1, 2, 3, 4, 6, 7, 8, 9)

	0	1
player 5 (i,j)	<div style="border: 1px solid black; padding: 5px; display: inline-block;">[-1, 0]</div> <div style="display: inline-block; vertical-align: middle;">↘ 1*</div>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">[-1, 0]</div> <div style="display: inline-block; vertical-align: middle;">↘ 1*</div>
1	1	<div style="border: 1px solid black; padding: 5px; display: inline-block;">[1, ..., 0]</div>

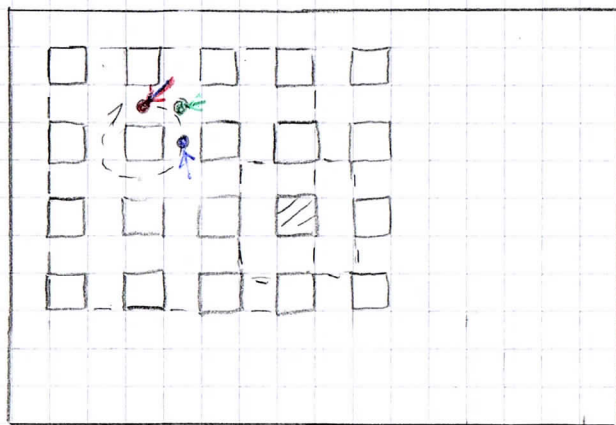
neighbor
D C

	D	C
player D	<div style="border: 1px solid black; padding: 5px; display: inline-block;">a</div> <div style="border: 1px solid black; padding: 5px; display: inline-block;">1</div>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">b</div> <div style="border: 1px solid black; padding: 5px; display: inline-block;">1</div>
player C	<div style="border: 1px solid black; padding: 5px; display: inline-block;">1</div>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">c</div>

see, GUI

Fig. 5a: CALCULATION OF PAYOFF FUNCTION

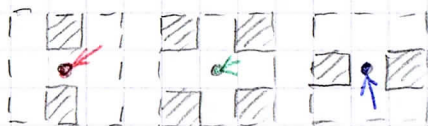
3/6



- # control cell is OFF
- # Algorithm to calculate
- # value of payoff function

A central cell is OFF

The set of correct neighbors: 3 correct neighborhoods



P1: 8 8 8
P2: 8 8 8

The set of incorrect neighbors

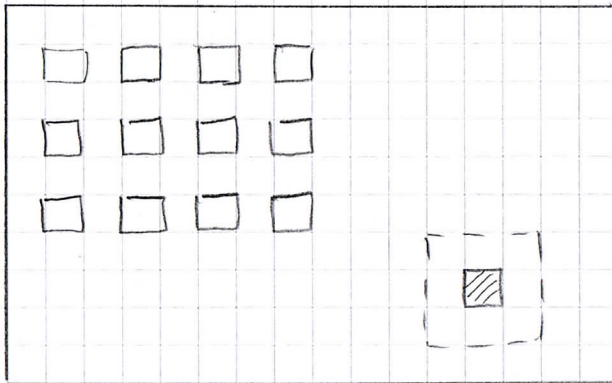


P1: 0
P2: -8

P1: $a=b=c=0$
P2: $a=b=c=-1$

Fig. 5b: CALCULATION OF PAYOFF FUNCTION

4/6



central cell is **ON**
 # Algorithm to calculate
 # value of payoff function

A central cell is **ON**

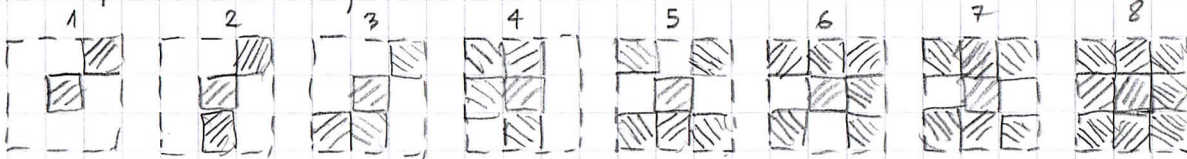
The set of correct neighborhoods: 1



P1:
P2:

8
8

The set of incorrect neighborhoods



P1
P2

7
6

6
4

P1: $a=b=c=0$
 P2: $a=b=c=-1$

Simulation parameters

(M+1) rows
 (N+1) cols
 p-init-C

sharing ☐

Competition-type

☐ loc-prop-sel
☒ loc-toun

p-state-mut

p-strat-mut

☐ seed

num-of-iter

num-of-exper

payoff function
neighbor

player

	D	C
D	<input type="text" value="a"/> <input type="text" value="1"/>	<input type="text" value="b"/> <input type="text" value="1"/>
C	<input type="text" value="1"/>	<input type="text" value="c"/>

START

Strategien

all-C
 all-D
 k-D
 k-C
 k-DC

☐ k-const

☒ k-var

Species:

☒ type 1 ☐
☐ type 2 ☐
☐ type 3 ☐

☐ debug

☐ random initpl

☐ deterministic initpl

Read CA states

Read CA strategies

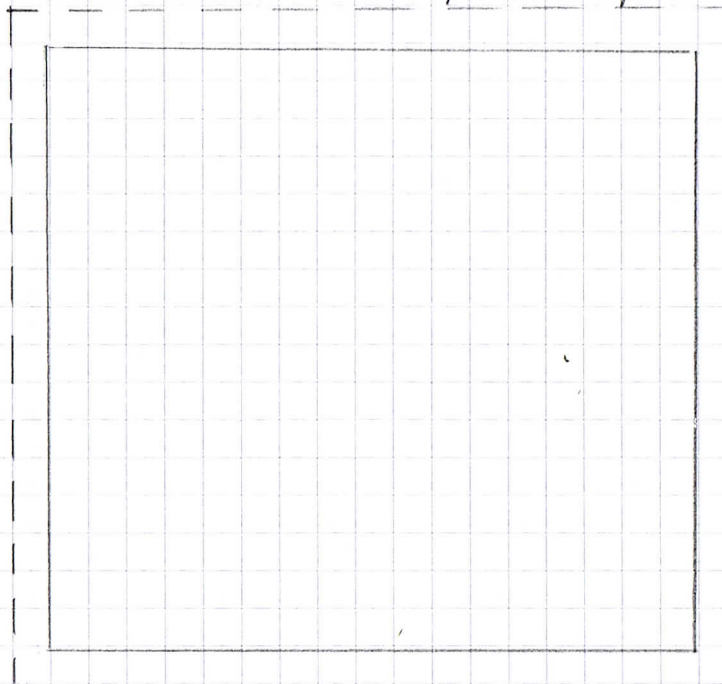
States

Strategies

KD strateg

KC strateg

KDC strateg



iter stop

Start animation

Stop animation

Save statistics

Save pictures

GUT

STRATEGY MUTATION

Mutated strategy → Resulting strategy

all-C → all-D

all-D → all-C

k-D → 0.5 all-D
0.5 all-C

k-C → 0.25 all-D
0.25 all-C
0.25 k-D
0.25 k-DC

k-DC → 0.25 all-D
0.25 all-C
0.25 k-D
0.25 k-C