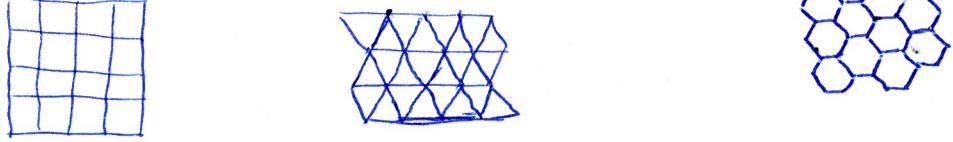


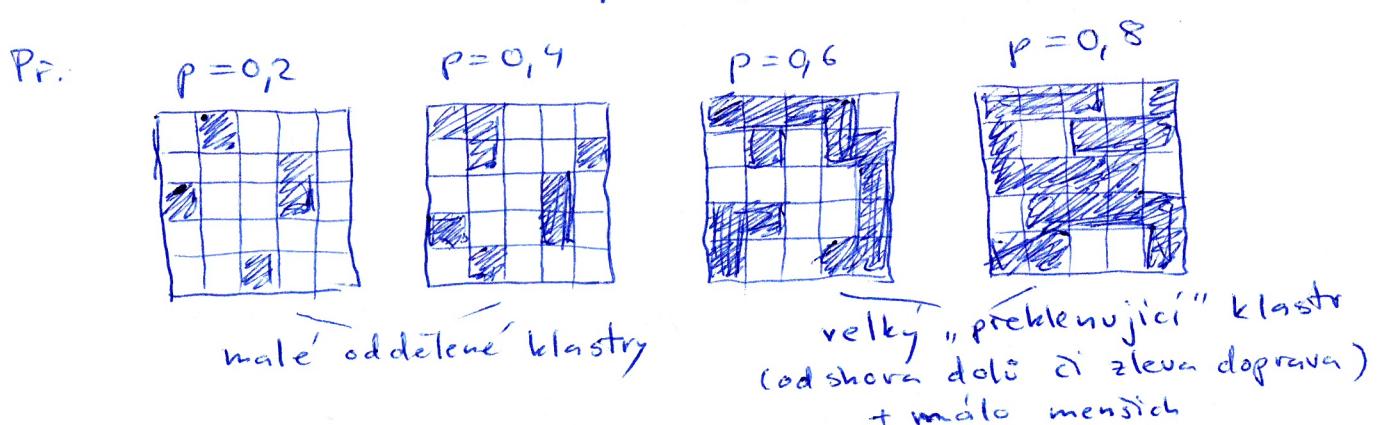
Perkolace - MC simulace

- teorie perkolace studuje počet kластů a jejich vlastnosti (velikost, trvaní apod.) v „náhodných“ strukturách a jevy, které se v nich vyskytují
 - obvykle modelujeme určitý reálný systém (např. pourovlnné látky, les, polymery apod.) na specifické mřížce v d-dimensionálním prostoru např. v dvourozměrném prostoru můžeme povázit čtvercovou trojúhelníkovou nebo hexagonální mřížku
 
 - nebo ve třírozměrném kubickou, dianantovou, bcc, fcc atd.
 - v závislosti na modelovaném systému můžeme studovat mřížky s obsazenými pozicemi (site percolation) nebo mřížky s obsazenými vazbami (bond percolation)

např.


 - počet obsazených míst (nebo vazeb) typicky závisí na určitém stanoveném paramétru systému (např. koncentraci usazovanej látky v plynu ci kapalině, hustoty stran v lese, teplotě atd.) a není známe, která konkrétní místa (vazby) jsou obsazena
 \Rightarrow v teorii perkolace se takové systémy modelují jako náhodně vytvořené struktury, přičemž pravděpodobnost p , že je místo (nebo vazba) obsazeno, je základním parametrem v každé simulaci na místo např. koncentrace

- tato pravděpodobnost se obvykle falešnou v základních modelech a simulacích) uvažuje jako nezávislá pro každé místo (vazbu), ale obecně může záviset na tom, zda jsou či nejsou obsazena sousední místa
- teorie perkolace pak studuje distribuci různě velkých klastrov v těchto mřížkách a zvláště pak pravděpodobnost výskytu nekonečných (spanning =? překlenujících) klastrov pro konečné mřížky v závislosti na pravděpodobnosti p



- pro každý typ mřížky existuje kritická hodnota pravděpodobnosti obsazení jednoho místa (případně jiného parametru jako koncentrace atd.) obvykle nazývaná (perkolaci) práh (threshold) p_c , při kterém dochází k přechodu od malých klastrov pro $p < p_c$ k velkým a nakonec překlenujícím klastrom pro $p > p_c$
- v případě nekonečné mřížky se pro $p < p_c$ nekonečný klastr nevyskytuje, kdežto pro $p > p_c$ je prítomen vždy (jde o „fázový přechod“ - systém všeobecně změní chování)

Príklad: při obsazení pozic jsou např. prahy

$$p_c = 0,59275 \dots \text{ pro čtvercovou mřížku}$$

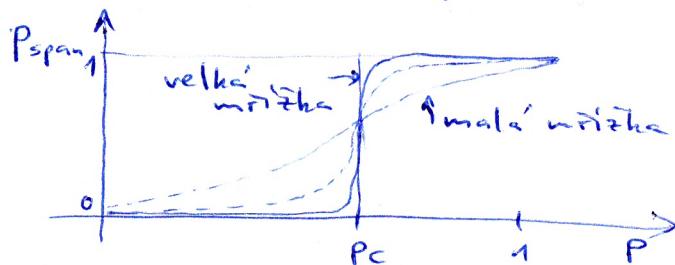
$$p_c = 0,5 \text{ pro trojúhelníkovou (pravoúhlou) mřížku}$$

$$p_c = 0,3117 \dots \text{ pro kubickou mřížku} \\ \text{atd.}$$

a při obsazení vazeb jsou jiné, např.

$$p_c = 0,5 \text{ pro čtvercovou mřížku}$$

- pro konečné mřížky je nenuova pravděpodobnost, že se překlenující klastr objeví pro libovolné $p > 0$, ovšem pro $p \ll p_c$ je velká malá
- když mřížku zvětšujeme, pravděpodobnost P_{span} , že se překlenoucí klastr objeví pro $p < p_c$, jde k nule a pro $p_c < p$ jde naopak k 1



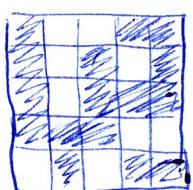
- pro některé mřížky lze prah p_c určit rigorózně matematicky (např. pro rámci dvourozměrných mřížek)
ovšem obecně je třeba je určit ze simulaci (alespoň přibližně)
např. tak, že pro různé velké mřížky určíme $P_{\text{span}}(p)$
a z průsečku těchto krivk lze odhadnout p_c
(viz notebook v Mathematici pro čtvercovou mřížku)
- k tomu je třeba pro lib. náhodně obsazovanou mřížku rozhodnout, zda se v ní vyskytuje překlenující klastr
- i když existují efektivnější způsoby zjištění tohoto výskytu, řekneme si, jak to udělat pomocí tzv. Hoshenova-Kopelmanova algoritmu, který umožňuje najít všechny klastry v jednom průchodu mřížkou
- tento algoritmus (publikován) ve Phys. Rev. B 14 (1976) 3438
ocísluje obsazená místa podle toho, ke kterému klastru dané místo patří
- jakmile máme klastry ocíslowane, můžeme jednoduše porovnání jejich císel v prvním a v posledním řádku (sloupcu) zjistit, zda se v mřížce nachází překlenující (spanning) klastr.

Hoshenov - Kopelmanov algoritmus pro odšlování klastru

- základní struktura:

- 1) systematicky projde všechny body mřížky
- 2) každému obsazenému místu přiřadí číslo klastru
 - bod podle již navštívených sousedních míst (obsazených)
 - nebo nové číslo, pokud sousedé nejsou obsazeni
- 3) vyřeší případné konflikty v odšlování,
 - pokud dvě sousední místa jsou již odšlovány různými čísly
 - k tomu se použije pomocné pole přiřazení odšlovacích klastru,
 - aby se nemusela předšlovat celá mřížka

Př.



jeden průchod
→

1		2	2	
1	3	2		
1	3	3	2	
1	1	1		
1		4	4	

přitom pomocné pole
priřazení čísel klastru bude

1	2	3	4
1	1	1	4

předšlování
(není nutné)
lze provést
přesněji
z H-K algoritmu

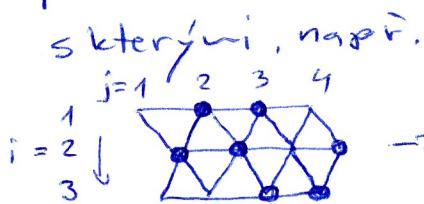
1		1	1
1	1	1	1
1	1	1	1
1	1	1	1
1		4	4

=> počet jedniček a čtyřek
dává velikost klastru
a protože se jedničky
nachází v první
a poslední řádku (sloupcu),
jde o překlenující klastru.

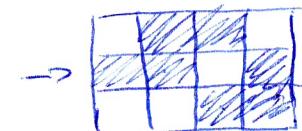
- Pozn.: 1) vždy přiřazujeme menší čísla větším při konfliktech
- 2) vždy je nutné změnit všechna stejná čísla v pomocném
poli při předšlování, např.
- | | | | | |
|---|---|---|---|---|
| ↑ | 1 | 2 | 3 | 4 |
| ↓ | 1 | 1 | 2 | 2 |
| ↔ | 1 | 2 | 1 | 4 |
- ← na zadní řádku
← po řádku
← pro sloupec
- 3) není potřeba udělat předšlování celé mřížky
po její projití - kompletní informace je v pomocném
poli, které používáme při počítání velikosti klastru
apod.

- modifikace pro jiné typy mřížek:

- i trojúhelníkové či hexagonální mřížky lze vložit do dvourozměrného pole, jen je pak potřeba dát pozor, která polička sousedí s kterými, např.



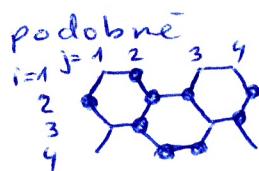
jediný kластr!



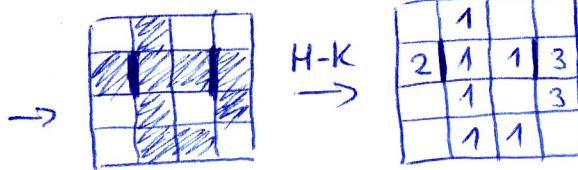
H-K algoritmus

přiřazení
1 → 1
2 → 1

- zde pole $(2,1)$ sousedi s $(1,2)$ proto rovnou přiřazuje 1 a $(2,2)$ sousedi s $(3,3)$!



ve skutečnosti 3 klastry



H-K

přiřazení
1 → 1
2 → 2
3 → 3

- zde $(2,1)$ nesousedi s $(2,2)$!
a také $(2,3)$ nesousedi s $(2,4)$!

- je nutné provést různé testy obsazenosti pro liché a sude řádky!

- počet sousedních polí ovlivňuje hodnotu p_c , čím více sousedů, tím je p_c menší
např. $p_c(\square) > p_c(\Delta) > p_c(\triangle)$