Simulations in Statistical Physics

Course for MSc physics students

Janos Török

Department of Theoretical Physics

October 28, 2014

Metropolis algorithm

(Metropoli-Rosenbluth-Rosenbluth-Teller-Teller=MR^{2}T^2 algorithm)

- Sequence of configurations using a Markov chain
- Configuration is generated from the previous one
- Transition probability: equilibrium probability
- Detailed balance:

$$P(x)W(x \to x') = P(x')W(x' \to x)$$

- Rewritten:

$$\frac{W(x \to x')}{W(x' \to x)} = \frac{P(x')}{P(x)} = e^{-\beta \Delta E}$$

- Only the ration of transition probabilities are fixed

Metropolis algorithm

(Metropoli-Rosenbluth-Rosenbluth-Teller-Teller=MR^{2}T^2 algorithm)

$$\frac{W(x \to x')}{W(x' \to x)} = \frac{P(x')}{P(x)} = e^{-\beta \Delta E}$$

- Metropolis:

$$W(x \to x') = \begin{cases} e^{-\beta \Delta E} & \text{if} \Delta E > 0 \\ 1 & \text{otherwise} \end{cases}$$

- Symmetric:

$$W(x \to x') = \frac{e^{-\beta \Delta E}}{1 + e^{-\beta \Delta E}}$$

Metropolis algorithm

Recipes:

- Choose an elementary step $x \to x'$
- Calculate ΔE
- Calculate $W(x \to x')$
- Generate random number $r \in [0, 1]$
- If $r < W(x \to x')$ then new state is x'; otherwise it remains x
- Increase time
- Measure what you want
- Restart

Metropolis algorithm, proposal probability

Transition probability:

$$W(x \to x') = g(x \to x')A(x \to x')$$

- $g(x \to x')$: proposal probability
  - Generally uniform
  - If different interactions are present then it must be incorporated
- $A(x \to x')$: acceptance probability
  - Metropolis
  - Symmetric

Metropolis, *proof*

State flow
Let $E > E'$:

- $x \to x'$

$$P(x)g(x \to x')A(x \to x') = P(x)$$

- $x' \to x$

$$P(x')g(x' \to x)A(x' \to x) = P(x')e^{-\beta\Delta E}$$

- In equilibrium they are equal:

$$\frac{P(x)}{P(x')} = e^{\beta\Delta E}$$

- What we wanted.

Do we need optimization?

- Correlation lenth ξ
- Characteristic time τ_{char}
- Dynamical exponent z

$$\tau_{\mathrm{char}} \propto \xi^z$$

- For 2d Ising model $z \simeq 2.17$
- Simulation time:

$$t_{\mathrm{CPU}} \sim L^{d+z}$$

We need more effective algorithms!

Multri-spin algorithm for 2d Ising model

History...

- Operations:
  - Check if neighbor is parallel: XOR
  - sum of antiparallel spins: sum of previous XOR
- Result: discrete energy difference can be 0, 1, 2, 3, 4

Metropolis	0	1	2	3	4
$\Delta E/J$	8	4	0	4	8
$W(x \to x')$	$\exp(-8\beta)$	$\exp(-4\beta)$	1	1	1

- (of course $W(x \to x')$ in array)
- 3 bit is enough to store result in 2d and 3d
- Use every fourth bit to store a spin.

Multri-spin algorithm for 2d Ising model

- Historical solution
  - Every fourth bit in the integer is a spin
  - We get `sizeof(int)/4` bits at once
  - Special bit order

NxN 8 bit integer L=N/2

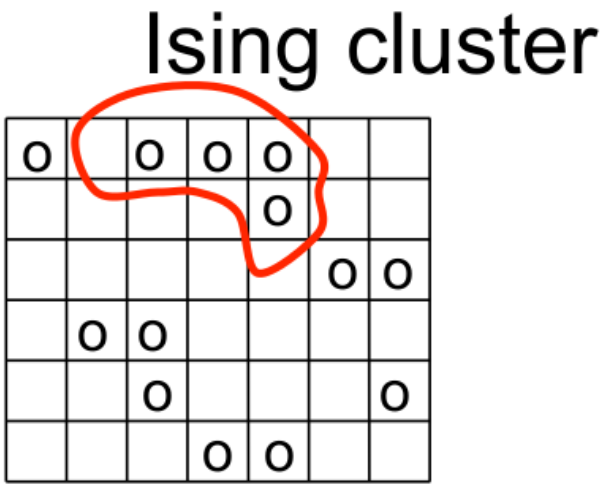
1	L+1	2	L+2	3	L+3
N+1	N+L+1	N+2	N+L+2	N+3	N+L+3
2N+1	2N+L+1	2N+2	2N+L+2	2N+3	N+L+3

Multri-spin algorithm for 2d Ising model

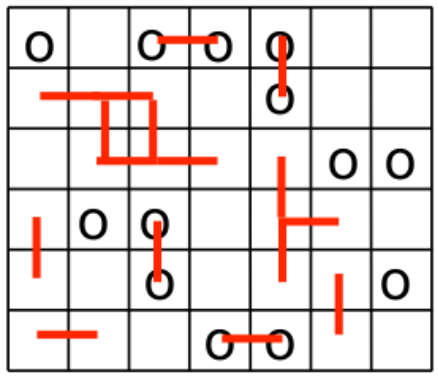
- Historical solution
  - Every fourth bit in the integer is a spin
  - We get `sizeof(int)/4` bits at once
  - Special bit order
  - Nowdays may even be slower as array operations are fast
- Use it for ensemble average
  - One member of the array contains the spin of one position
  - Multiple simulation instances
  - With Metropolis algorithm few random numbers are needed (at high T)
- Does not really matter only factors can be won, $t_{\mathrm{CPU}} \sim L^{d+z}$ still holds

Cluster algorithm

- Flip more spins together. How?
- The solution – based on an old relationship between the percolation and the Potts model – is that we consider the spin configuration as a correlated site percolation problem
- Ising cluster: a percolating cluster of parallel spins
- Ising droplets: a percolating subset of an Ising cluster
  $p_B = 1 - \exp(-2\beta J)$



Ising configuration



Ising „droplets”

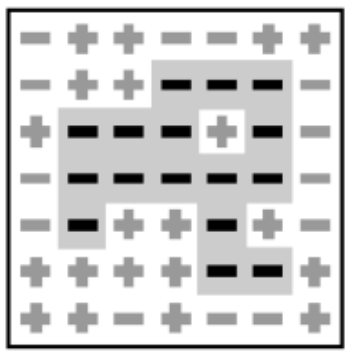
Swendsen-Wang algorithm

- Take an Ising configuration
- With probability $p_B = 1 - \exp(-2\beta J)$ make connection between *parallel* spins
- Identify the droplets by Hoshen-Kopelman algorithm
- Flip each droplet with probability: $1/2$ ($h = 0$)
- Repeat it over

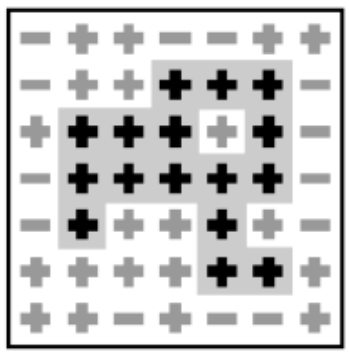
Wolff algorithm

1. Add a random spin to a list of active spins
2. Take a spin from the active list
3. Add each parallel neighboring (not yet visited) spin with probability $p_B = 1 - \exp(-2\beta J)$ to the list of active spins
4. If list of active spins is not empty go to 2.
5. Flip all active spins

A Wolff droplet (gray) before flipping



a



b

The new configuration The droplet contour is still shown, though the bonds are eliminated after flipping

Wolff algorithm *proof*

- Detailed balance:

$$P^{eq}(x)W(x \to x') = P^{eq}(x')W(x' \to x)$$

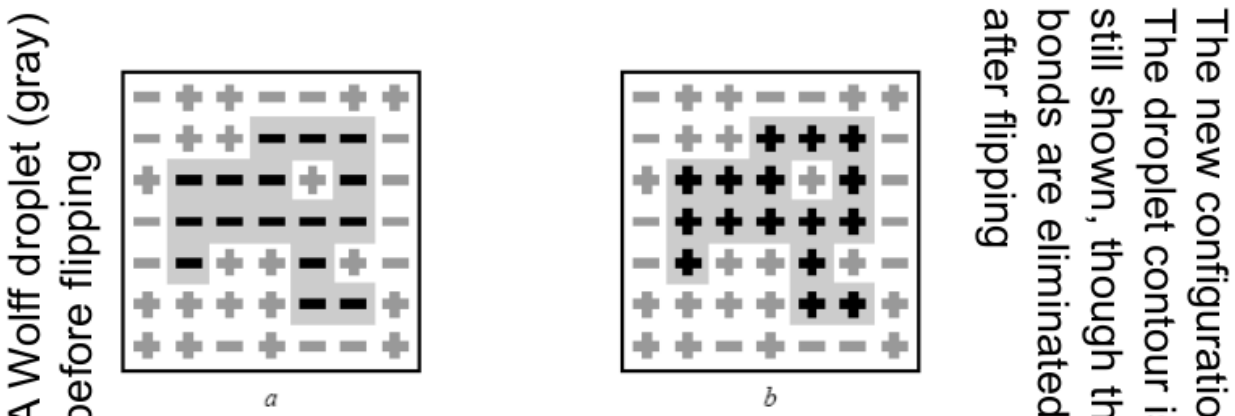
- Metropolis:

$$W(x \to x') = \min\left\{1, \frac{P^{eq}(x)}{P^{eq}(x')}\right\}$$

- Split W into acceptance A and proposal g probability

$$A(x \to x') = \min\left\{1, \frac{P^{eq}(x)g(x' \to x)}{P^{eq}(x')g(x \to x')}\right\}$$

Wolff algorithm *proof*



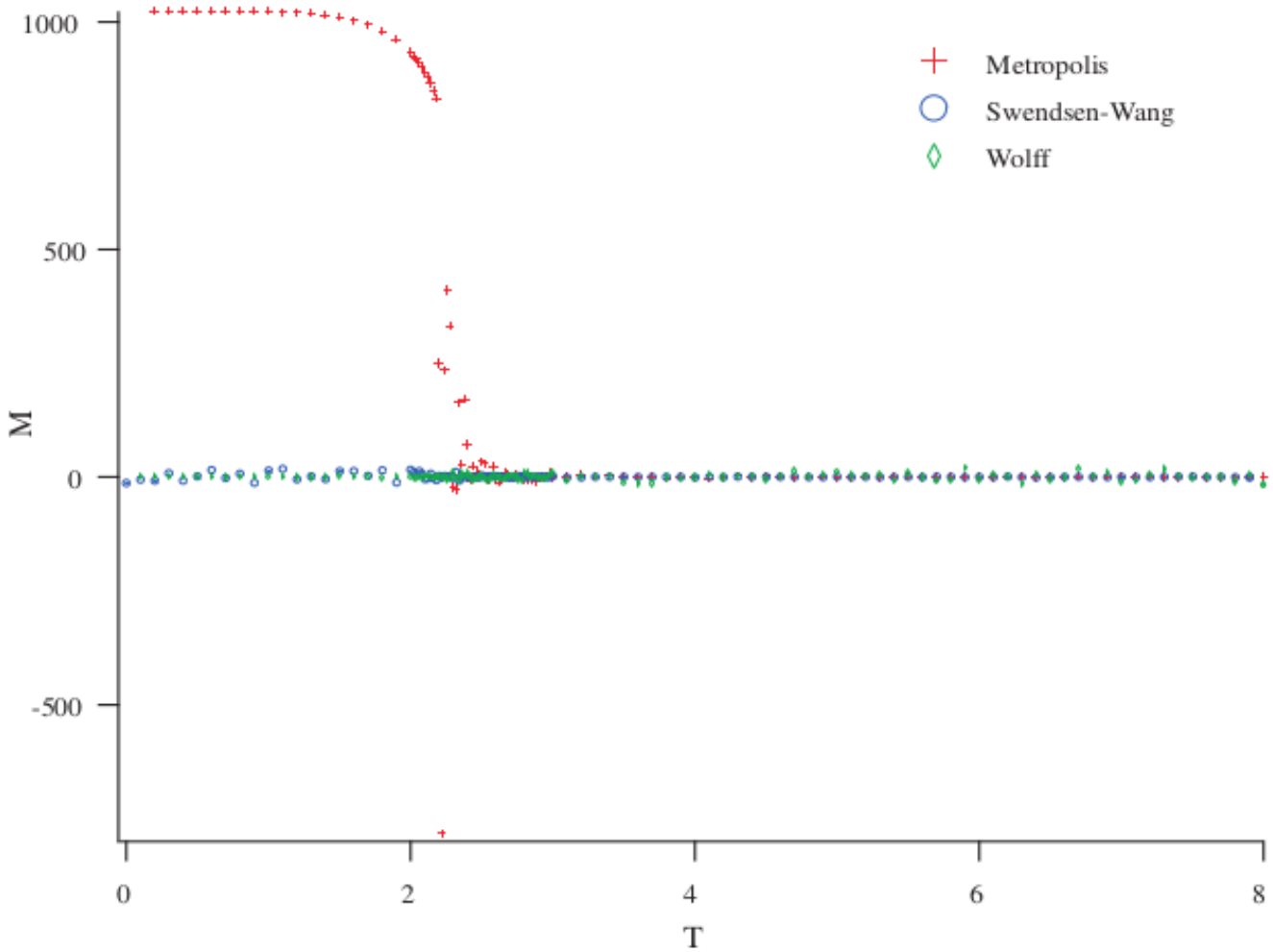
- On the boundary: n_{same} spins parallel and n_{diff} antiparallel.

$$A(x \to x') = \min\left\{1, \frac{e^{\beta J(n_{\text{diff}} - n_{\text{same}})}}{e^{\beta J(n_{\text{same}} - n_{\text{diff}})}} \frac{(1-p_B)^{n_{\text{diff}}}}{(1-p_B)^{n_{\text{same}}}}\right\}$$

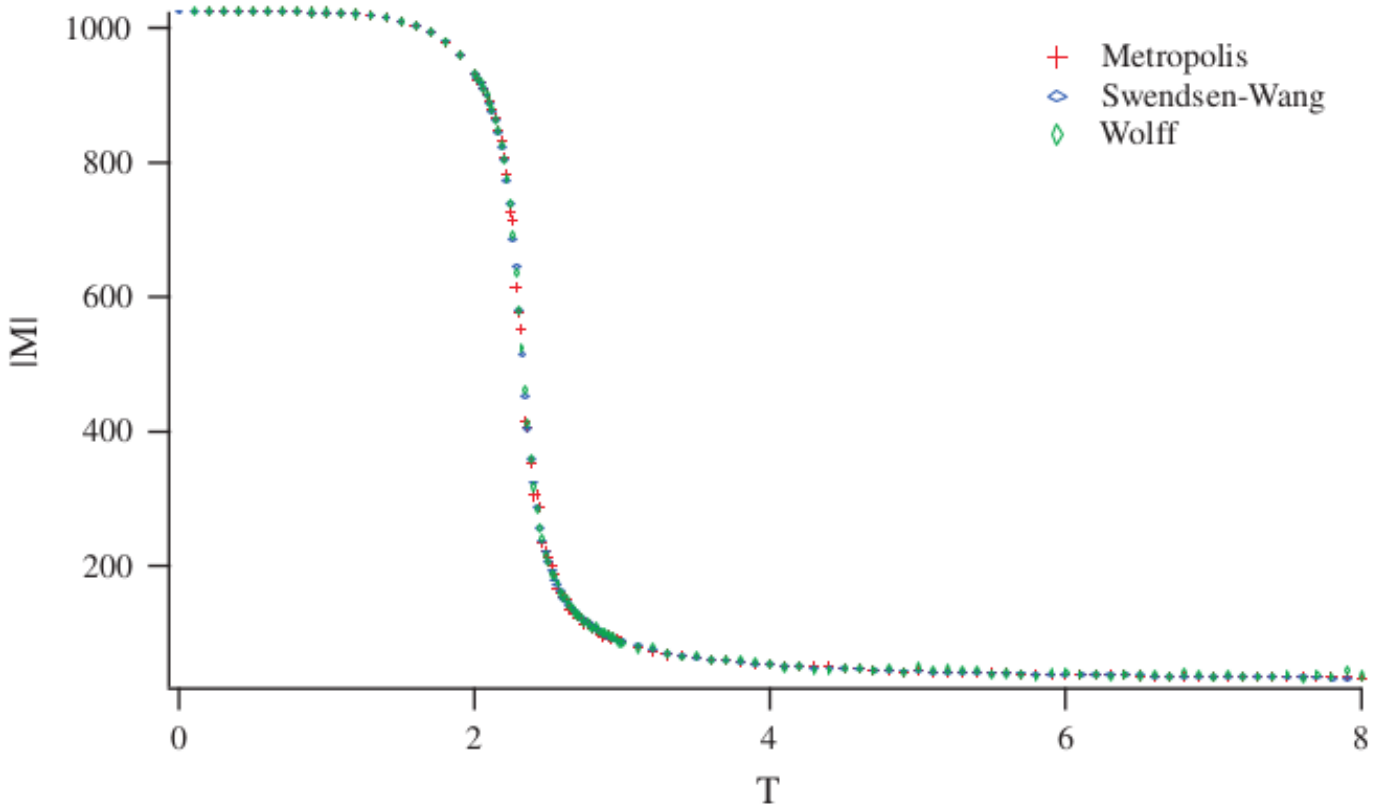
$$= \min\left\{1, \frac{e^{-2\beta J n_{\text{same}}}}{e^{-2\beta J n_{\text{diff}}}} \frac{(1-p_B)^{n_{\text{diff}}}}{(1-p_B)^{n_{\text{same}}}}\right\}$$

- It gives: $p_B = 1 - \exp(-2\beta J)$.

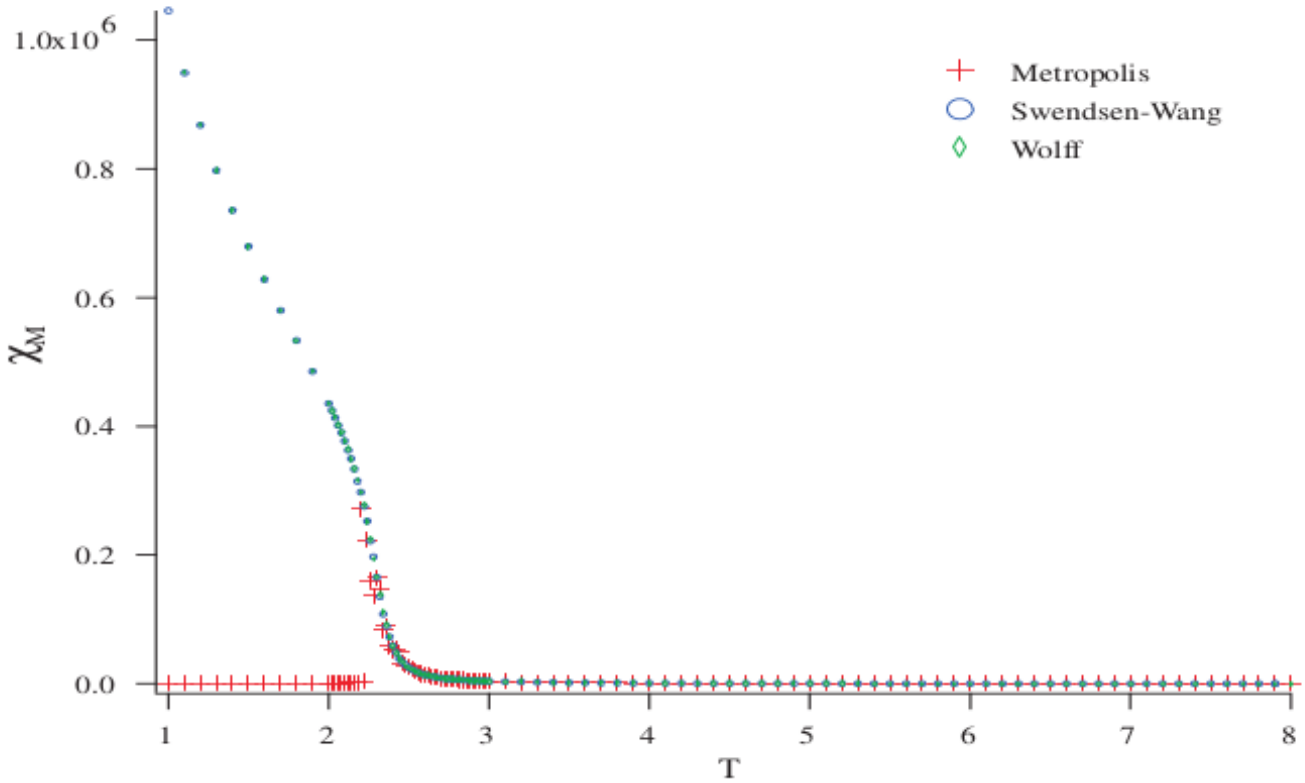
Comparison magnetization



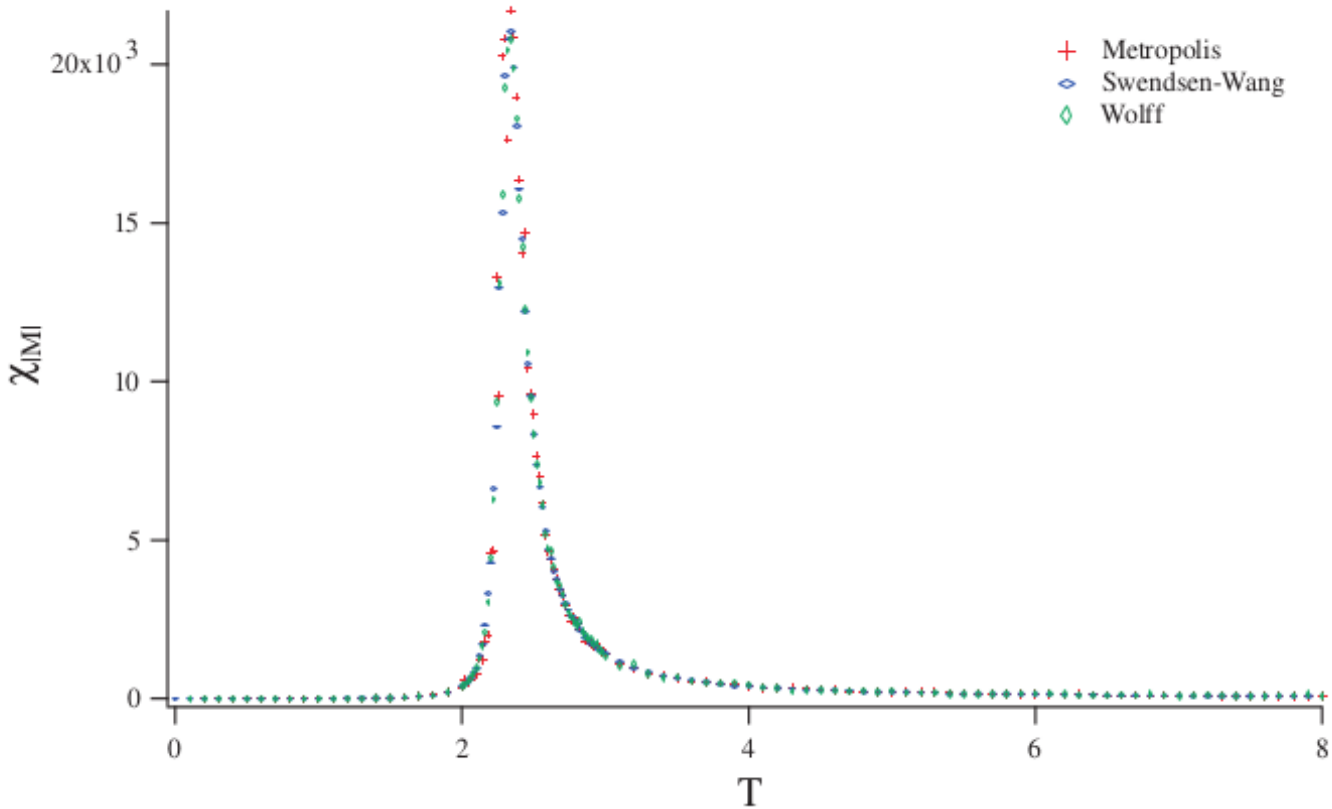
Comparison magnetization



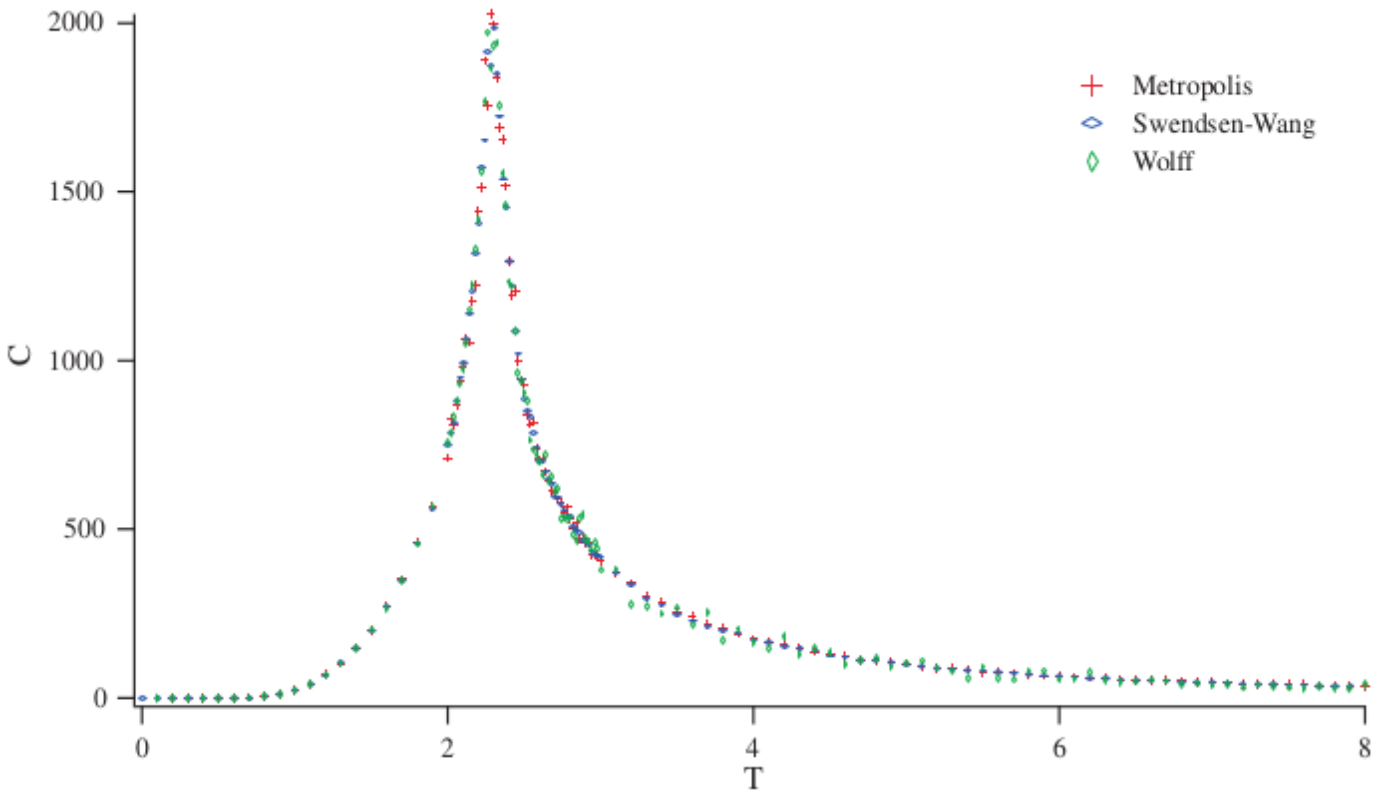
Comparison magnetization



Comparison magnetization



Comparison magnetization



Other ensembles

Microcanonical ensemble

- Daemon with bag with tolerance (both directions)
  - Pick a move, and calculate energy change
  - If energy change does not fit into bag reject it
  - Otherwise add energy change to bag
- In case of conservation the dynamic exponent z is larger!

Other ensembles

Conserved order parameter: Kawasaki dynamics

- Elementary step:
  - Exchange up-down spin pairs (can be anywhere) simultaneously
  - Apply Metropolis to net energy change!
  - Diffusive dynamics is more physical: pick neighboring spins
- In case of conservation the dynamic exponent z is larger!

Calculation of the entropy, free energy, etc.

- Equilibrium statistical physics: From F we can calculate everything
- In simulations F and S cannot be measured directly
- $F = E - TS$ so one of them is enough (E and T are known)
- Solution:
  Calculate the specific heat!

$$C = k_B T^2 \langle (\Delta E)^2 \rangle$$

- The energy fluctuations are measurable
- Since

$$C = T \frac{\partial S}{\partial T}$$

We have

$$S(T) = S(T_0) + \int_{T_0}^{T} \frac{C(T')}{T'} dT'$$

Calculation of the entropy, free energy, etc.

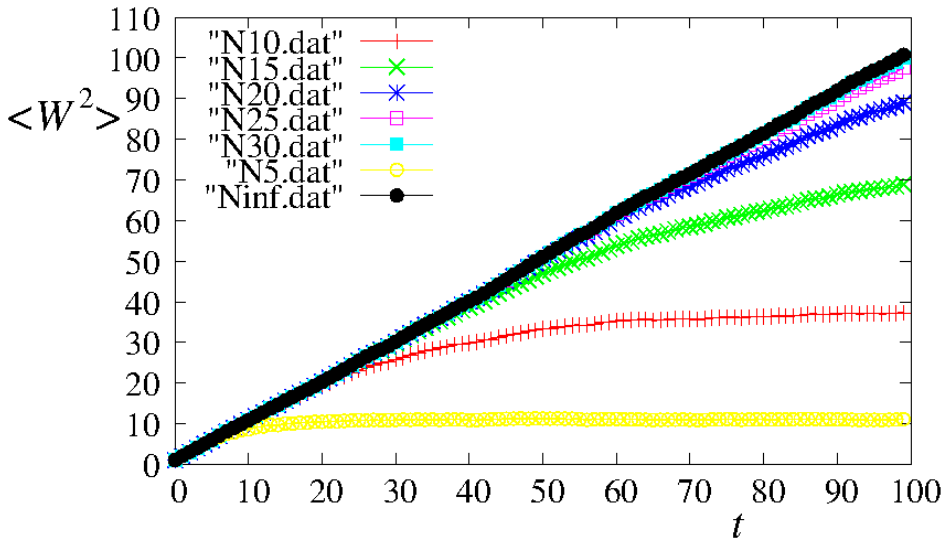
- In many cases derivate of the entropy is needed so $S(T_0)$ is not important in

$$S(T) = S(T_0) + \int_{T_0}^{T} \frac{C(T')}{T'} dT'$$

- From third law of thermodynamics: $S(T = 0) = 0$.

Diffusion

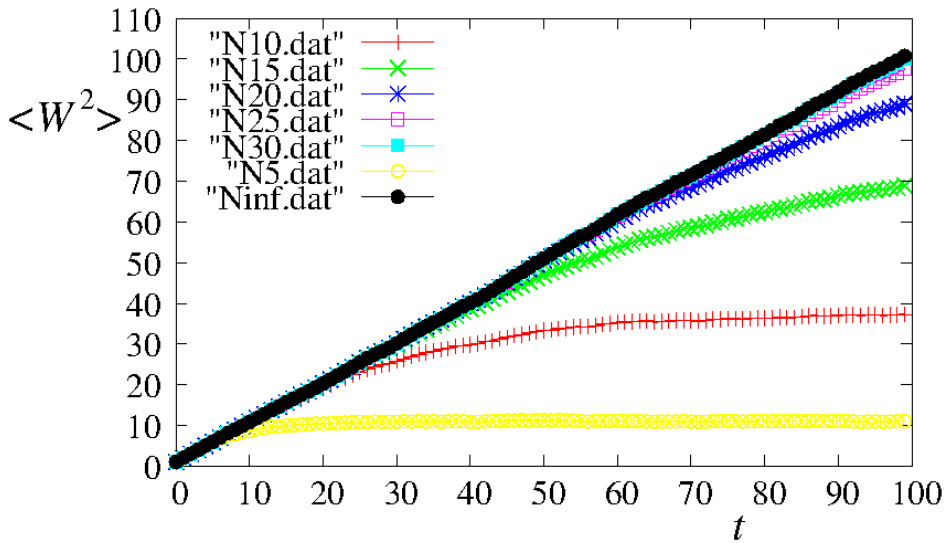
- On normal lattice exactly sovable
- Otherwise e.g. Monte Carlo kinetics. E.g. 1D
  - With probability $1/2 \rightarrow$ go right
  - With probability $1/2 \rightarrow$ go left
  - Be careful with boundary conditions



- Can easily be biased
- Can be simulated on spurious lattices, e.g. Parcolation clusters

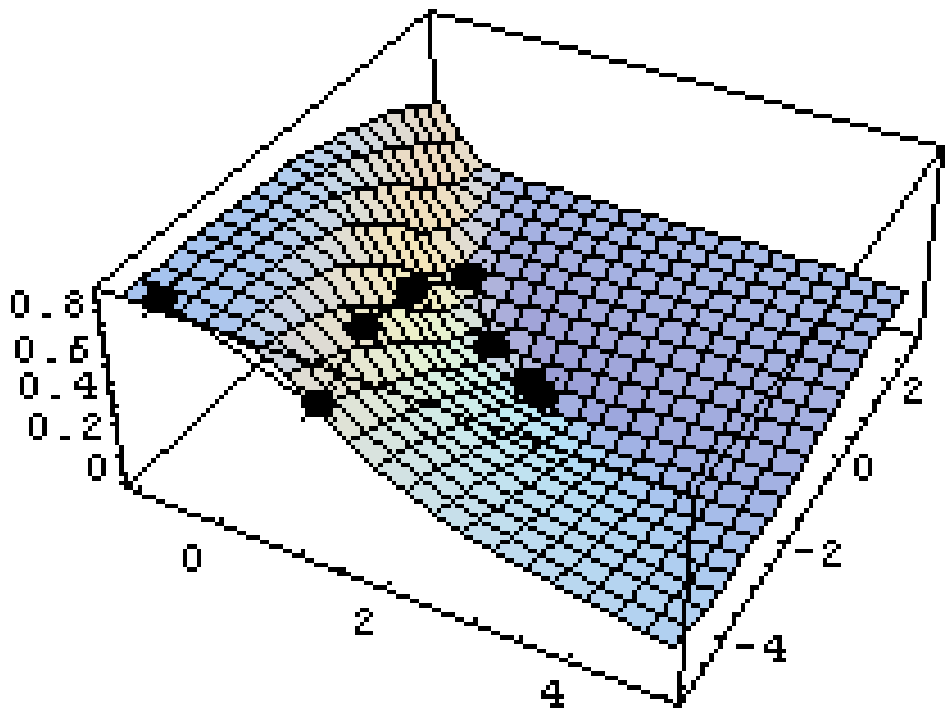
Diffusion

- Solution for diffusion on finite lattice:
- Count steps in both directions
- The net move is $W = n_+ - n_0$
- Use ensemble average
- Plot $\langle W^2 \rangle$ vs. t



Optimization

- General problem of finding the ground state
- Phase-space:
- Arbitrary number of dimensions
- Methods:
  - Steepest Descent
  - Stimulated Annealing
  - Genetic algorithm



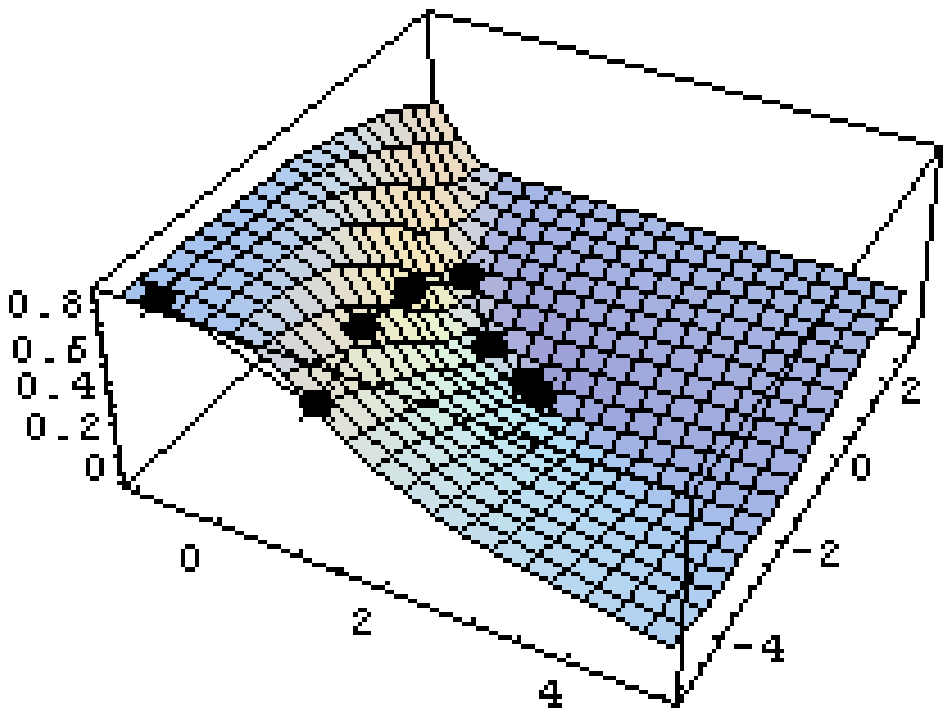
Gradient based optimization

- Given $f(\mathbf{x})$, with $\mathbf{x} = \{x_1, x_2, \ldots x_n\}$
- Gradient $\nabla f(\mathbf{x}) \equiv \mathbf{g}(\mathbf{x}) = \{\partial_1 f, \partial_2 f, \ldots \partial_n f\}$
- Second order partial derivatives: square symmetric matrix called the *Hessian matrix*:

$$\nabla^2 f(\mathbf{x}) \equiv H(\mathbf{x}) \equiv \begin{pmatrix} \partial_1\partial_1 f & \ldots & \partial_1\partial_n f \\ \vdots & \ddots & \vdots \\ \partial_1\partial_n f & \ldots & \partial_n\partial_n f \end{pmatrix}$$

General Gradient Algorithm

1. Test for convergence
2. Compute a search direction
3. Compute a step length
4. Update **x**



Steepest descent algorithm

1. Start from $\mathbf{x}_0$
2. Compute $\mathbf{g}(\mathbf{x}_k) \equiv \nabla f(\mathbf{x}_k)$. If $||\mathbf{g}(\mathbf{x}_k)|| \leq \varepsilon_g$ then stop, otherwise, compute normalized search direction $\mathbf{p}_k = -\mathbf{g}(\mathbf{x}_k)/||\mathbf{g}(\mathbf{x}_k)||$
3. Compute α_k such that $f(\mathbf{x}_k + \alpha \mathbf{p}_k)$ is minimized
4. New point: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{p}_k$
5. Test for $|f(\mathbf{x}_{k+1} - f(\mathbf{x}_k))| \leq \varepsilon_a + \varepsilon_r |f(\mathbf{x}_k)|$ and stop if fulfilled in two successive iterations, otherwise go to 2.

Conjugate Gradient Method

1. Start from $\mathbf{x}_0$
2. Compute $\mathbf{g}(\mathbf{x}_k) \equiv \nabla f(\mathbf{x}_k)$. If $||\mathbf{g}(\mathbf{x}_k)|| \leq \varepsilon_g$ then stop, otherwise Go to 5
3. Compute $\mathbf{g}(\mathbf{x}_k) \equiv \nabla f(\mathbf{x}_k)$. If $||\mathbf{g}(\mathbf{x}_k)|| \leq \varepsilon_g$ then stop, otherwise continue
4. Compute the new conjugate gradient direction $\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1}$, where

$$\beta = \left(\frac{||\mathbf{g}_k||}{||\mathbf{g}_{k-1}||} \right)^2$$

5. Compute α_k such that $f(\mathbf{x}_k + \alpha \mathbf{p}_k)$ is minimized
6. New point: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{p}_k$
7. Test for $|f(\mathbf{x}_{k+1} - f(\mathbf{x}_k))| \leq \varepsilon_a + \varepsilon_r |f(\mathbf{x}_k)|$ and stop if fulfilled in two successive iterations, otherwise go to 3.

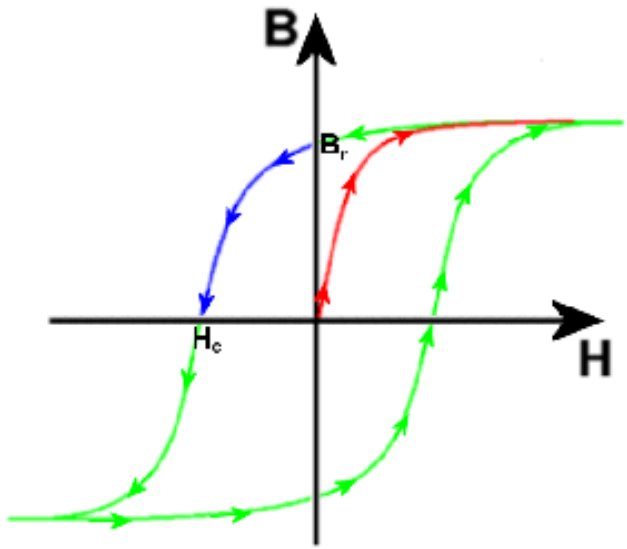
Modified Newton's method

Second order method

1. Start from $\mathbf{x}_0$
2. Compute $\mathbf{g}(\mathbf{x}_k) \equiv \nabla f(\mathbf{x}_k)$. If $||\mathbf{g}(\mathbf{x}_k)|| \leq \varepsilon_g$ then stop, otherwise, continue
3. Compute $H(\mathbf{x}_k) \equiv \nabla^2 f(\mathbf{x}_k)$ and the search direction $\mathbf{p}_k = -H^{-1}\mathbf{g}_k$
4. Compute α_k such that $f(\mathbf{x}_k + \alpha \mathbf{p}_k)$ is minimized
5. New point: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{p}_k$
6. Go to 2.

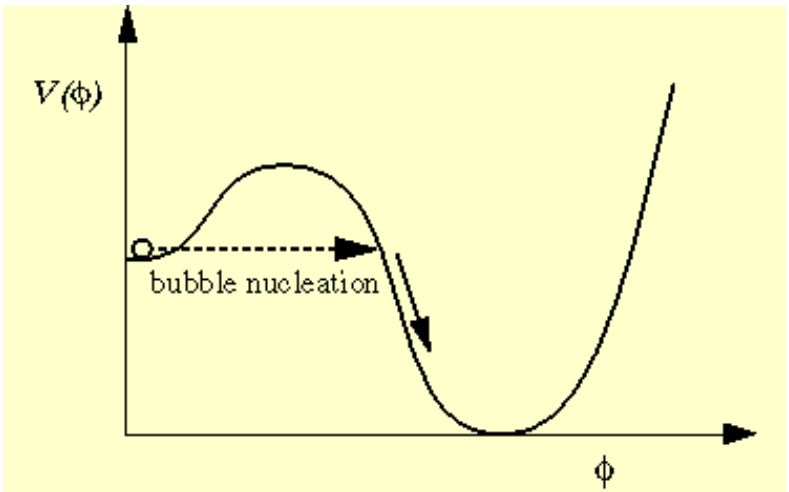
Metastability

- At first order transitions the correlation length remains finite.
- The mechanism of the first order transition is usually nucleation, which is related to metastability.
- Examples can be observed at hysteresis or undercooling, overheating, over-compessing etc.



Nucleation

- There is a competition between the bulk free energy of the droplet and its surface energy
- There is a critical nucleus size above which the transition is very rapid.
- However, such a critical nucleus has to be created by spontaneous fluctuations – which takes (sometimes enormously long) time.

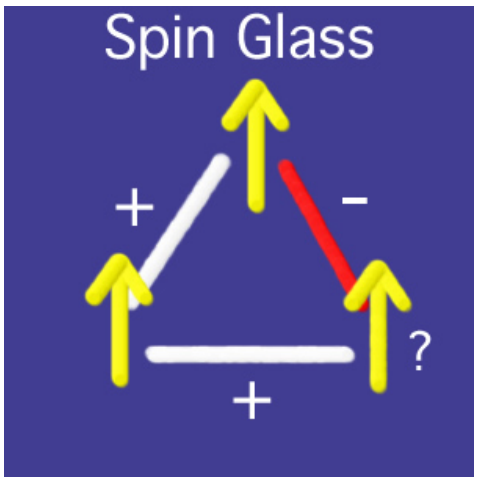


Glassy behavior, frustration

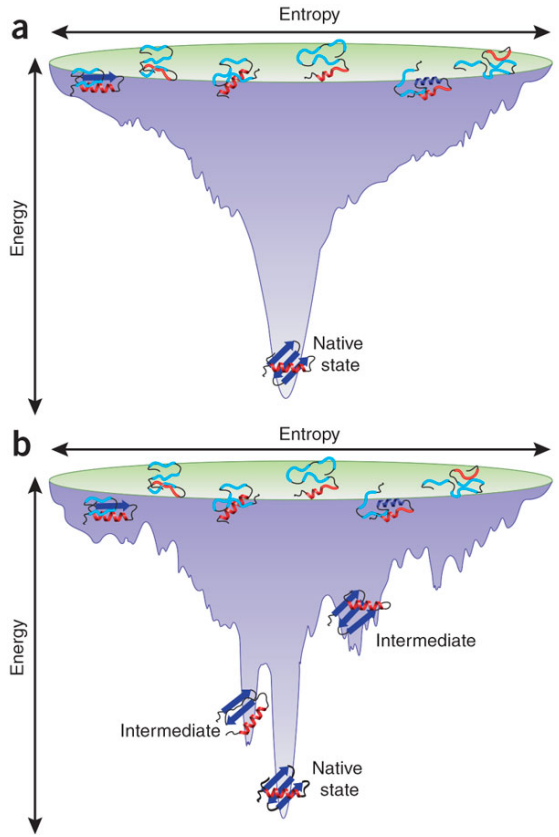
- Model glass: spin-glass:

$$H = -\frac{1}{2}\sum_{\langle i,j\rangle} J_{ij} S_i S_j$$

- where J_{ij} are random quenched variables with 0 mean (e.g. $\pm J$ with probability half)



Rugged energy landscape.



Rugged energy landscape

- Typical example NP-complete problems:
  - Traveling salesman
  - Graph partitioning
  - Spin-glass
- No full optimization is possible (do we need it?)
- Very good minimas can be obtained by optimization
  - Simulated annealing
  - Genetic algorithm

Simulated annealing

- Cool down the system slowly
- Warm up and down if needed, if the system quenched into a local minimum
- One needs a Hamiltonian and an elementary move
- Traveling salesman
  - Path length
  - Exchange two cities in the path
- Use Metropolis simulated annealing. ($T \sim$ alcohol)

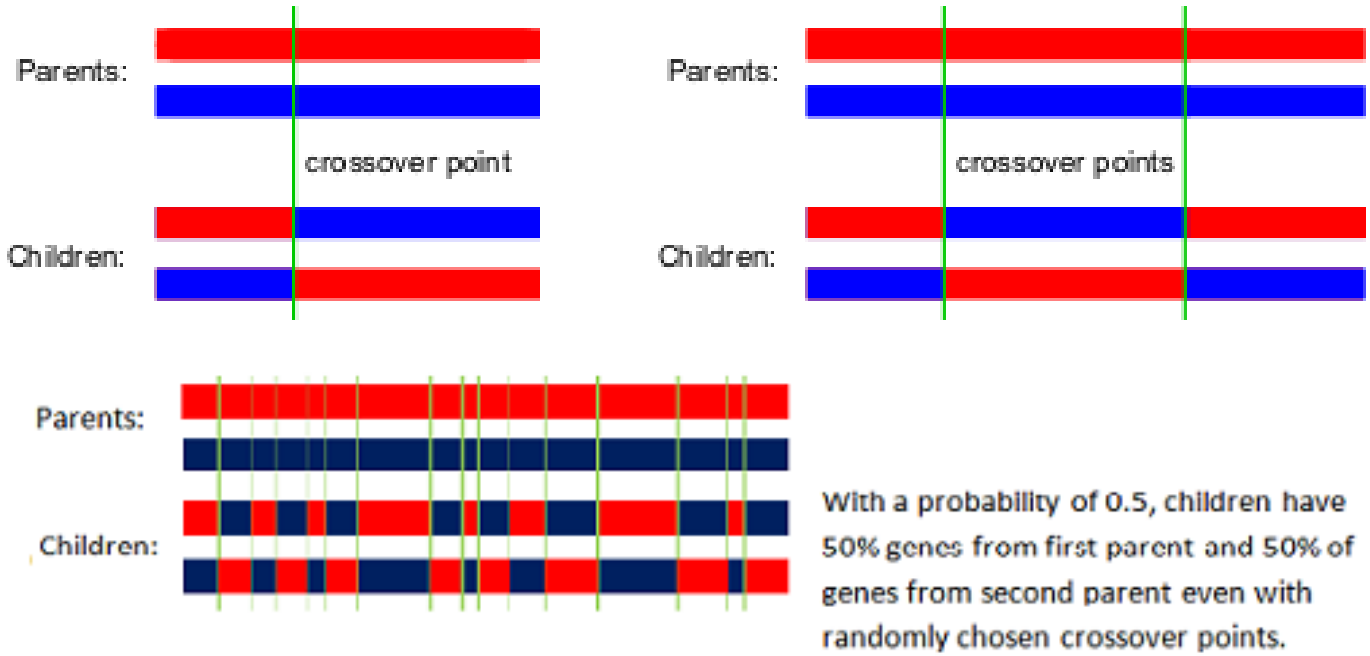
Demo movie

Genetic algorithm

- Learn from nature
- Let the fittest to survive
  - Fitness function, e.g. energy, length, etc.
- Combine different strategies
- State is represented by a vector (genetic code or genotype)
  - Phasespace, city order, neural network parameters, etc.
- Offsprings have two parents with shared genetic code
- Mutations
- Those who are not fit enough die out
  - Keep the number of agents fixed

Genetic algorithm: Reproduction

- Two parents and two children



Genetic algorithm terminology

- Chromosome: Carrier of the genetic representation
- Gene: Smallest units in the chromosome with individual meaning
- Parents: Pair of chromosomes, wich produce offsprings
- Population: Set of chromosomes from which the parents are selected. Its size should be larger than the length of the chromosome
- Selection principle: The way parents are selected (random, elitistic)
- Crossover: Recombination of the genes of the parents by mixing
- Crossover rate: The rate by which crossover takes place ($\sim$90%)
- Mutatation: Random change of genes
- Mutation rate: The rate by which mutation takes place ($\sim$1%)
- Generation: The pool after one sweep.

Genetic algorithm terminology