Problemy odwrotne w geofizyce (L5)

Wojciech Dębski

debski@igf.edu.pl

www.igf.edu.pl/~ debski

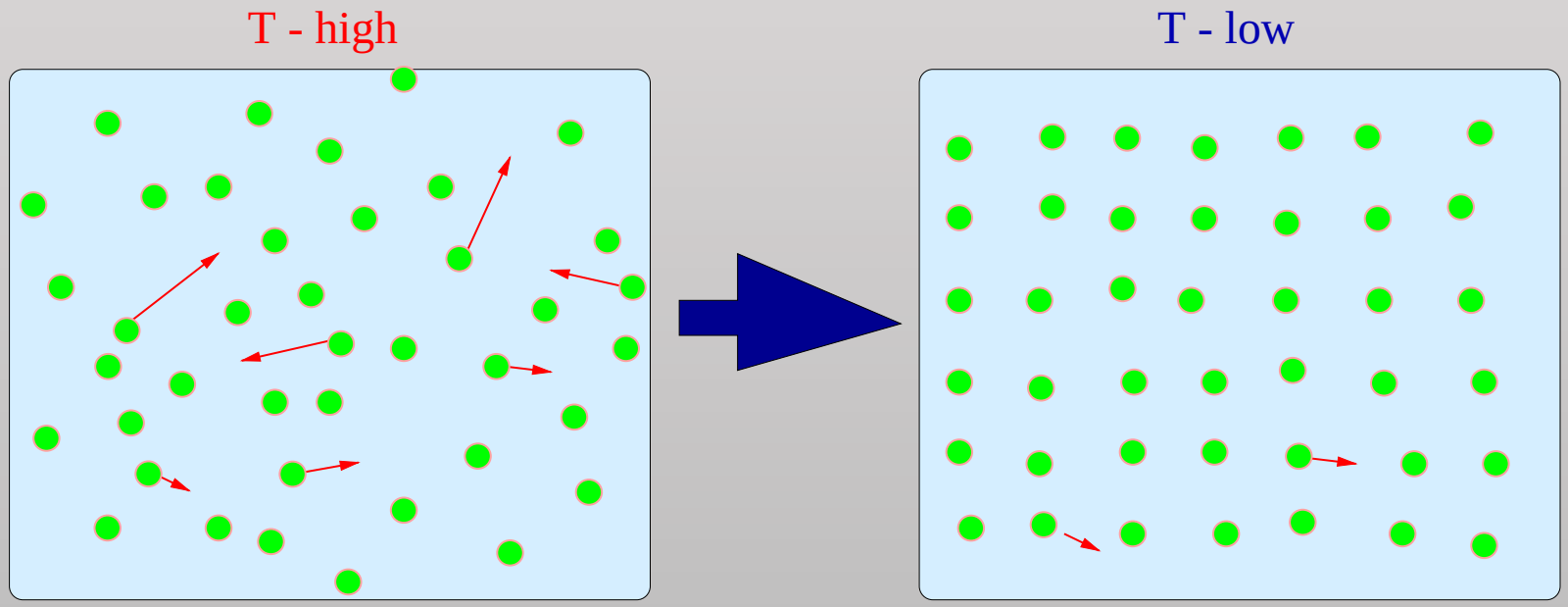
Plan wykładu

- Symulowane wyrzażanie - cd.
  - Podstawy fizyczne
  - Opis statystyczny
  - Optymalizacja a krystalizacja - analogie
  - Próbkowanie ważnościowe (ang. importance sampling)
- Technika łańcuchów Markowa (ang. *Markov Chain Monte Carlo*)
  - procesy stochastyczne
  - algorytm Metropolisa
  - algorytmy błądzenia przypadkowego
- Algorytmy typu Simulated Annealing
  - klasyczny Simulated Annealing
  - algorytm *Very Fast Simulated Annealing*
  - Jak używać algorytmu SA - “ściąga”

Proces schładzania

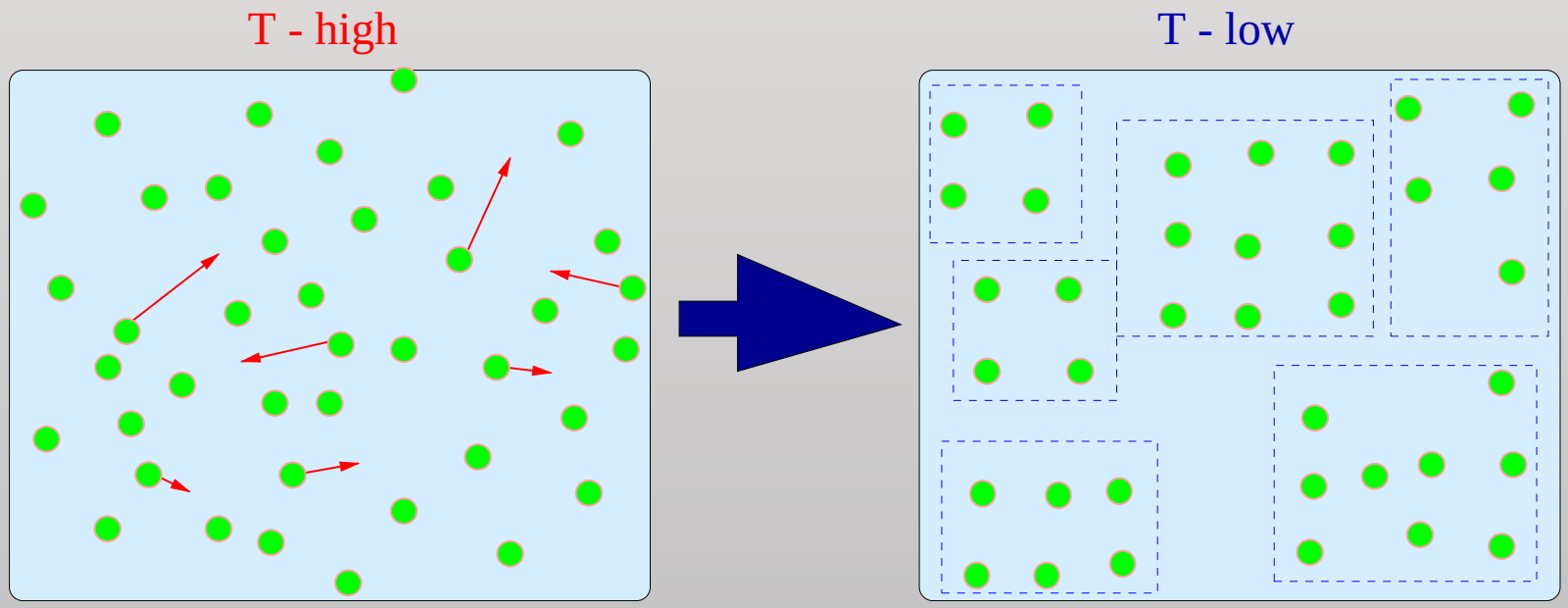
schładzanie powolne

Proces schładzania - system zdąża do stanu o minimum energii wewnętrznej
- ruchy termiczne zostają “zamrożone”



Proces schładzania

szybkie schładzanie



Układ fizyczny - opis statystyczny

- Interesują nas na ogół *makroskopiczne* wielkości: temperatura, gęstość, lepkość, itp.
- Wiele różnych *mikroskopowych* stanów odpowiada tej samemej wielkości makroskopowej
- Równowaga termodynamiczna: stan makroskopowy nie zmienia się w czasie jednakże stany mikroskopowy - tak

Thermodynamika równowagowa

Rozkład Boltzmana (kanoniczny)

$$p(C) = \frac{1}{Z} exp\left(-\frac{E(C)}{kT}\right)$$

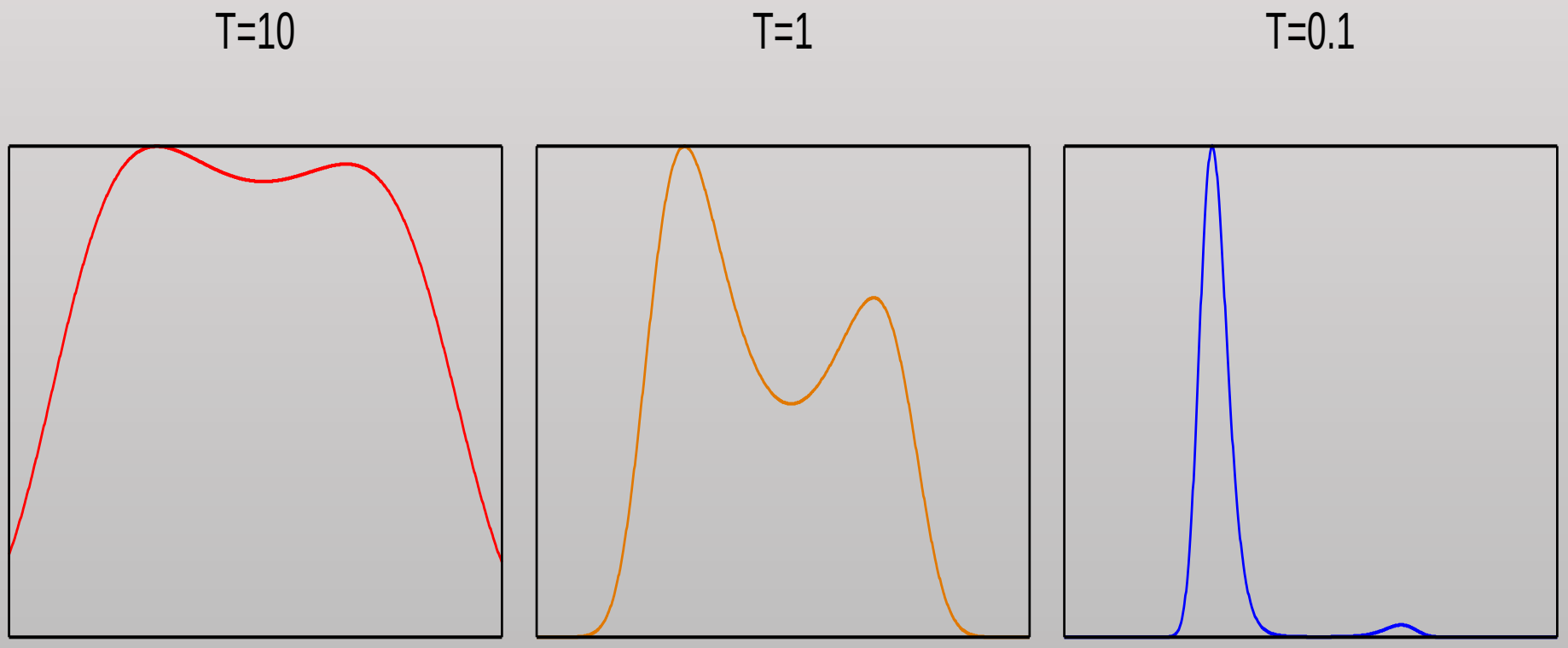
E - energy of a given state
T - temperature

$$Z(T, M, ...) = \sum_{C} exp\left(-\frac{E_i}{kT}\right)$$

prwdopodobieństwo realizacji stanu makroskopowego przez dany stan mikroskopowy C

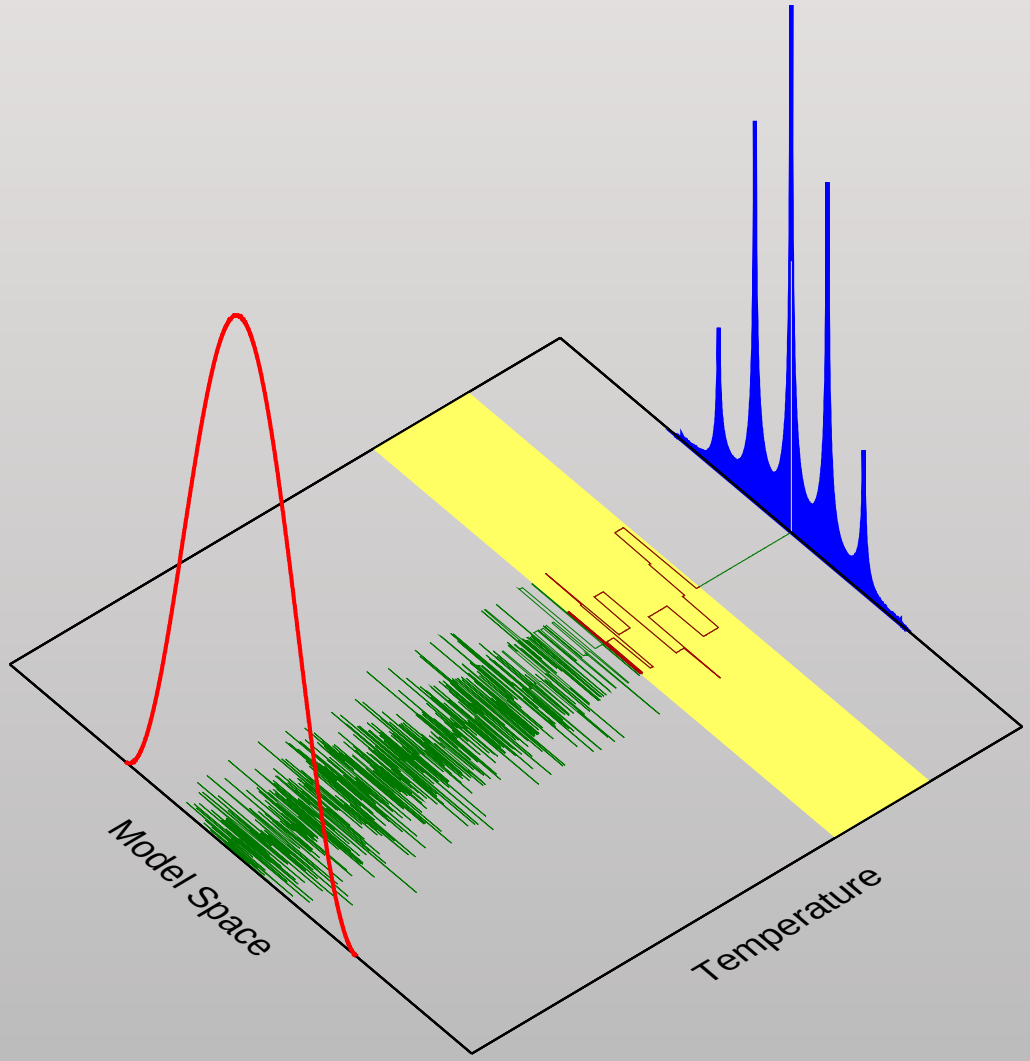
Proces schładzania

ewolucja rozkładu Boltzmana



Proces schładzania:

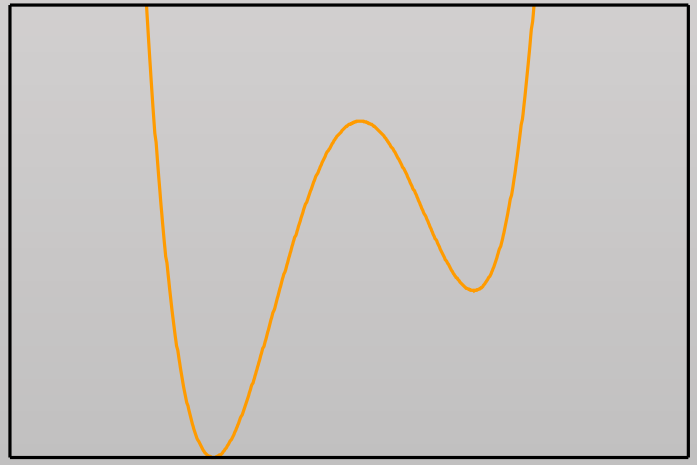
krystalizacja

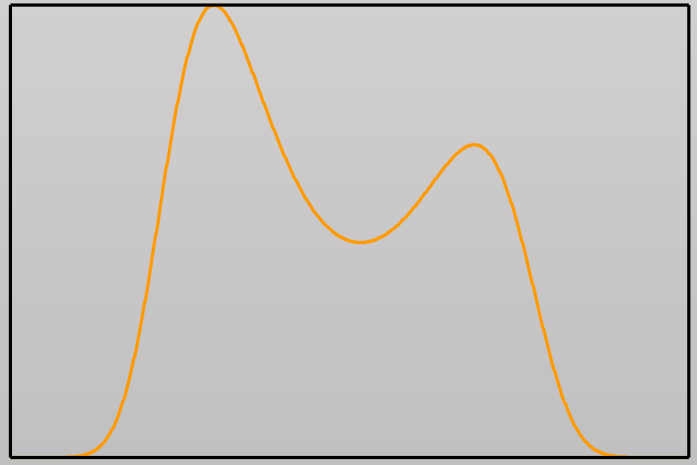


Symulowane schładzanie - optymalizacja

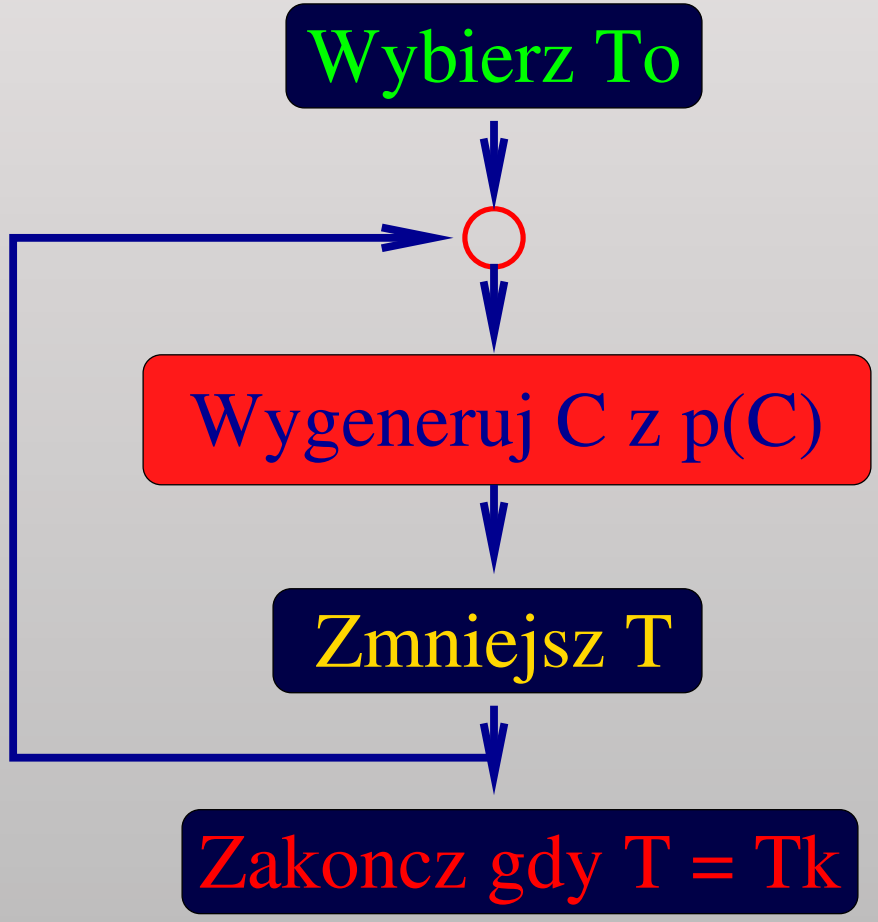
Szukamy absolutnego minimum dodatniej funkcji $S(\mathbf{m}) > 0$

$$S(\mathbf{m}) \Longrightarrow e^{-\frac{S(\mathbf{m})}{T}}$$

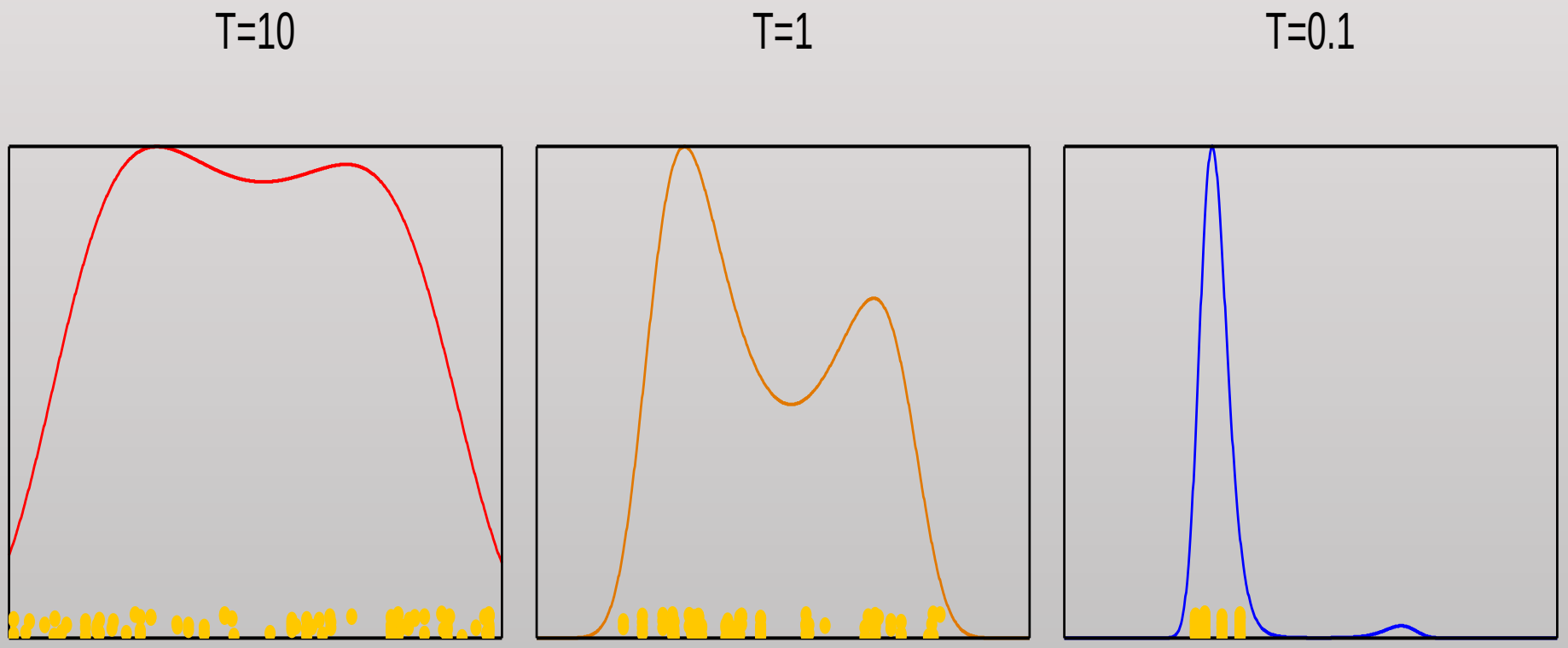




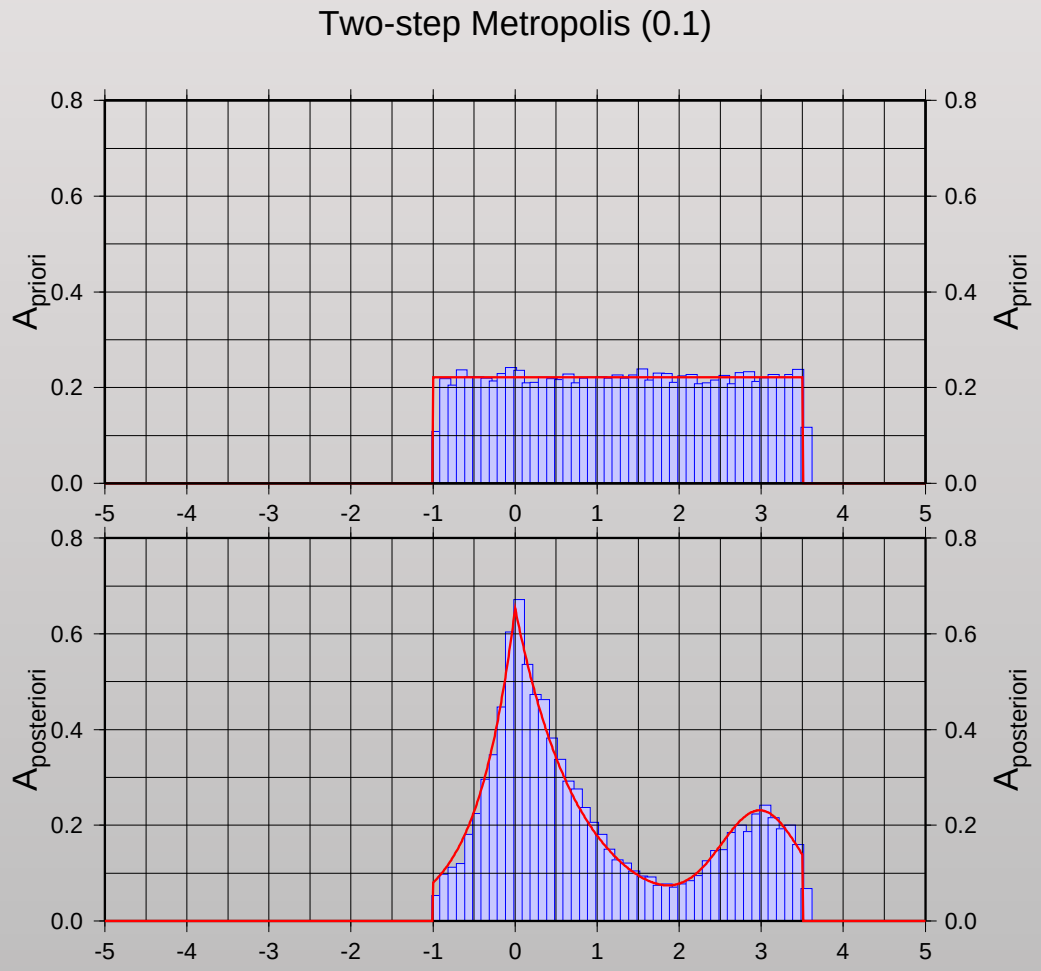
Algorytmy typu Simulated Annealing



Generowane próbki



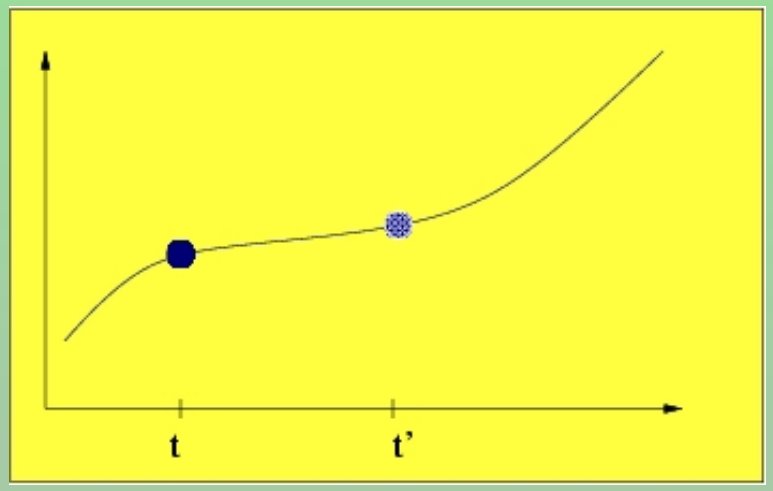
ang. *Importance Sampling*



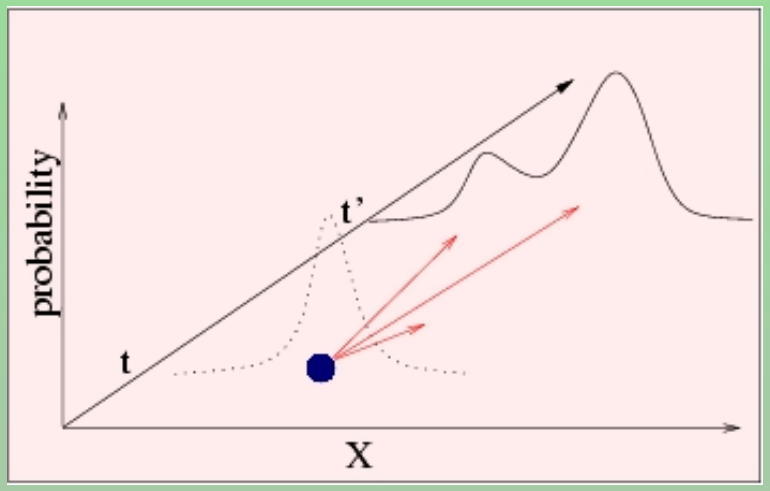
Procesy dynamiczne

(z krótką pamięcią)

deterministyczne



stochastyczne



$$X(t+dt) = X(t) + f(X(t),t)dt \qquad P(x,t) = \sum_{x'} P(x',t')K(t,x;t',x')$$

Łańcuchy Markowa

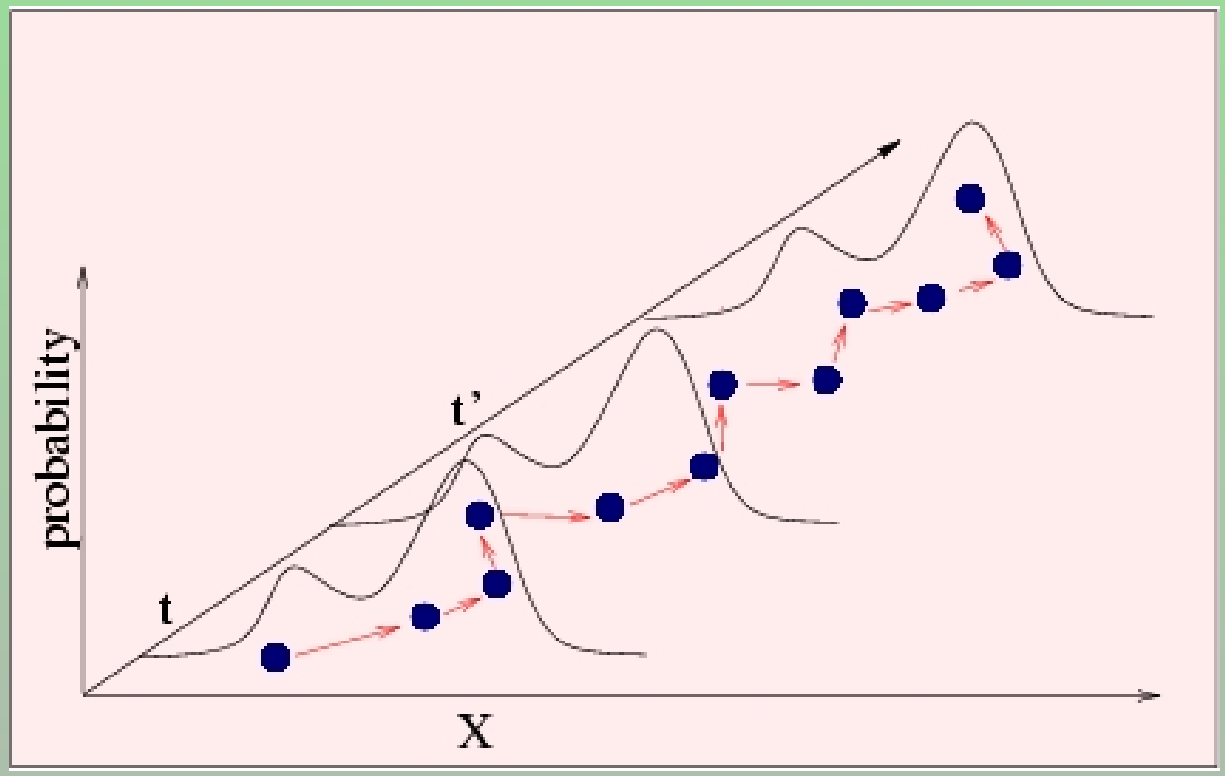
- dyskretny czas: $t_1, t_2, \cdots$
- proces stchastyczny z "krótką" pamięcią $K(t_i, x; t_{i-1}, x_{i-1})$
- proces ergodyczny:
  - stacjonarny $K(t, x; t', x') = K(x, x')$
  - nieredukowalny: (graf nieredukowalny)
  - aperiodyczny
- Istnieje rozkład stacjonarny: $\pi(x') = \pi(x)$

$$\pi(x) = \int_{x'} \pi(x') K(x; x') dx'$$

Stacjonarny łańcuch Markowa

$$p(x_i) = \sum_k p(x_k)\, K(x_i; x_k)$$

$$p(x) = \int_{x'} \pi(x') K(x; x') dx'$$

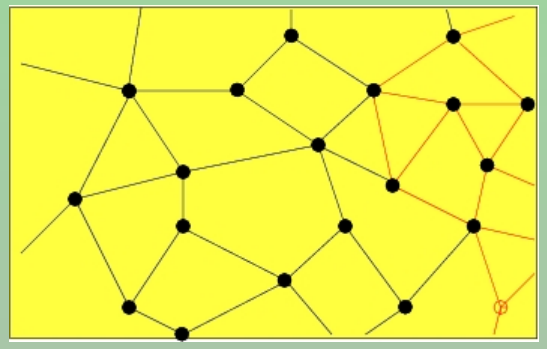


Realizacja łańcucha Markowa

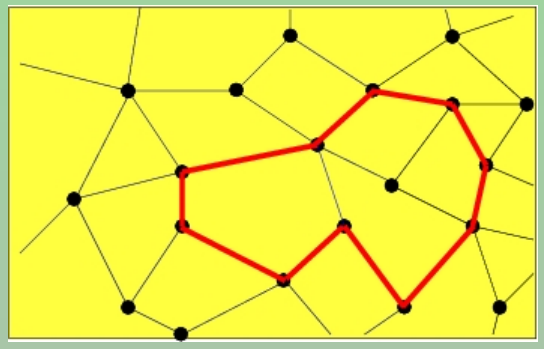
Błądzenie przypadkowe (ang. random walk

$$x_i \Longrightarrow x_{i+1} = x: \quad p_{x_i \to x_{i+1}} = K(x_{i+1}|x_i)$$

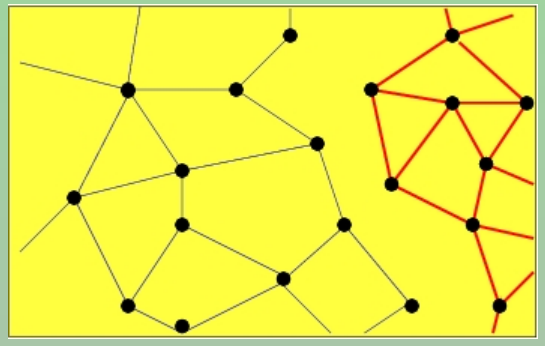
ergodyczny



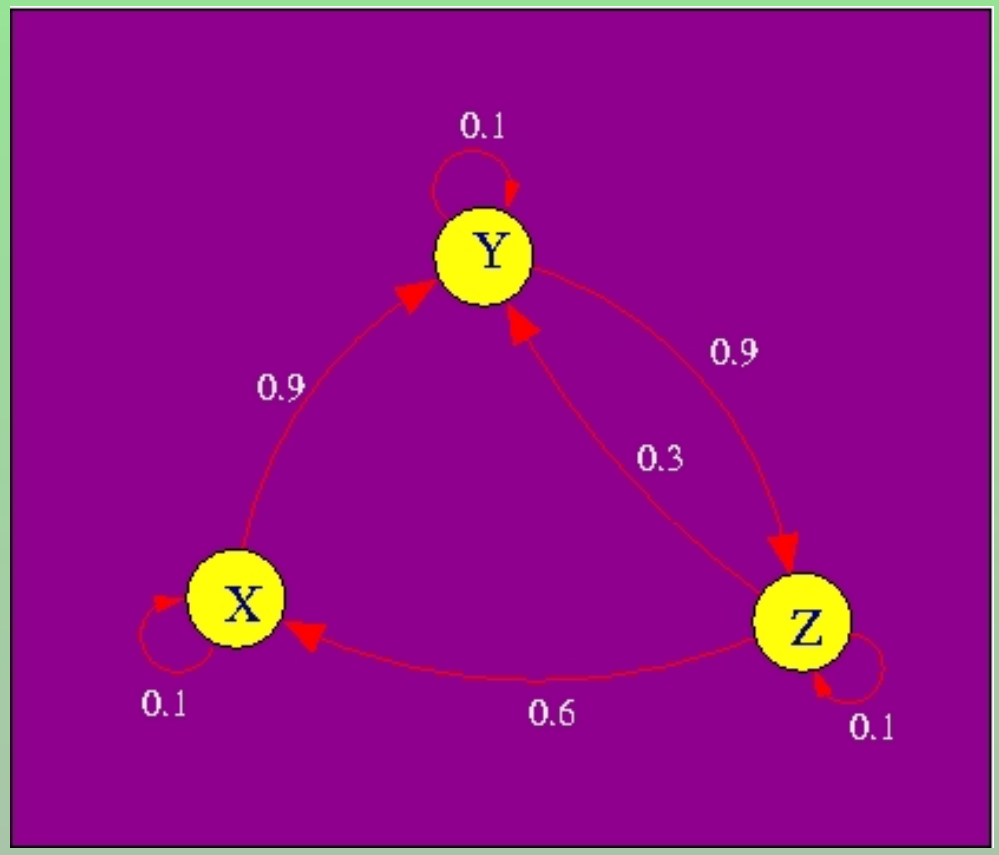
periodyczny



redukowalny



Algorytm przykład



Algorytm przykład

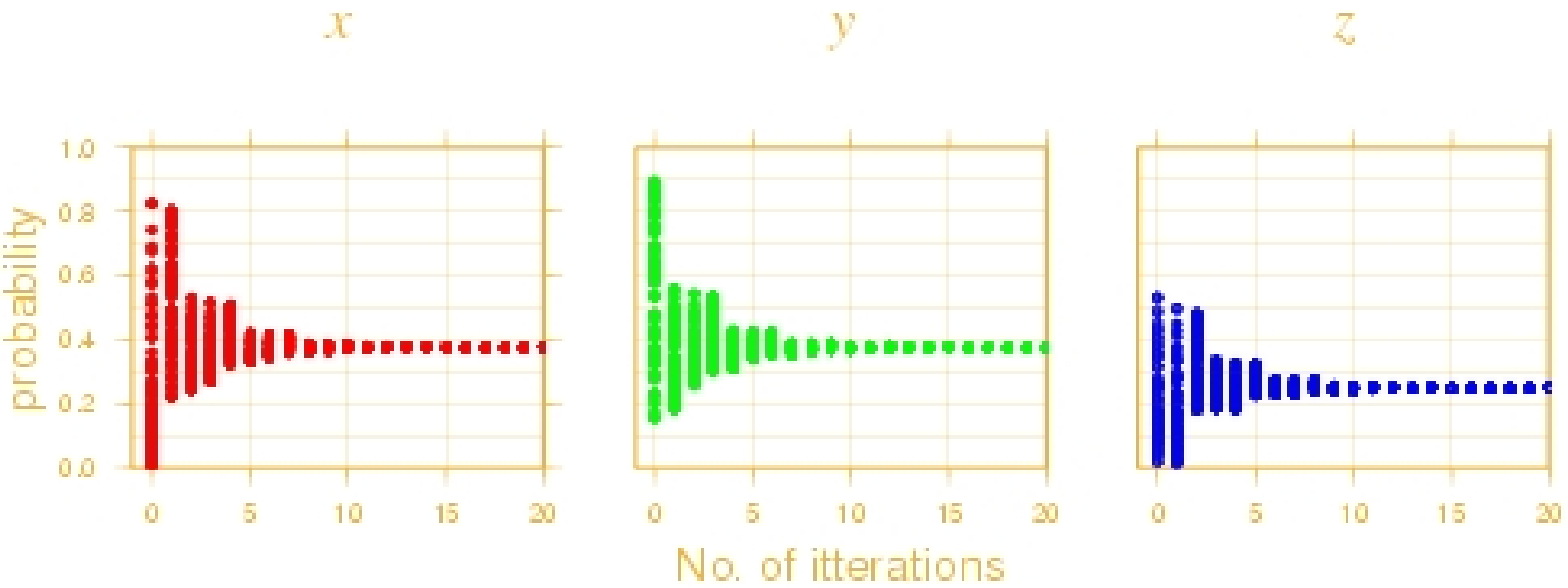
$$\mathbf{p}_i = (p_x, p_y, p_z)$$

Równanie ewolucji

$$\mathbf{p}_{i+1} = \mathbf{p}_i \cdot \begin{pmatrix} 0.1 & 0.9 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.3 & 0.1 \end{pmatrix}$$

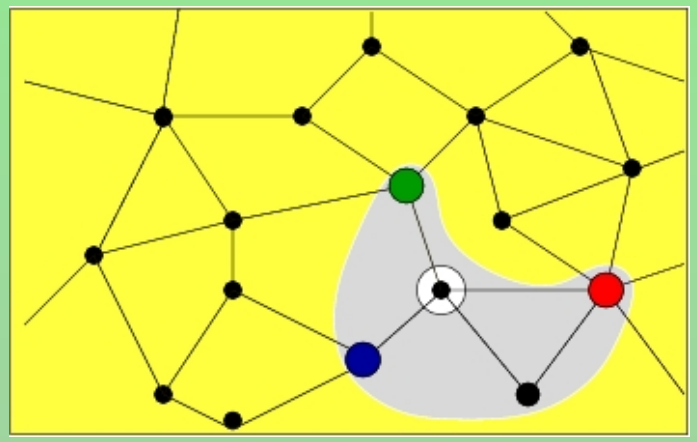
$\mathbf{p}_{i\to\infty} =???$

Algorytm przykład



Określenie procesu

1.



2.

$$K = \begin{pmatrix} K_{11} & \cdots & K_{1N} \\ K_{21} & \cdots & K_{2N} \\ \vdots & \vdots & \vdots \\ K_{M1} & \cdots & K_{MN} \end{pmatrix}$$

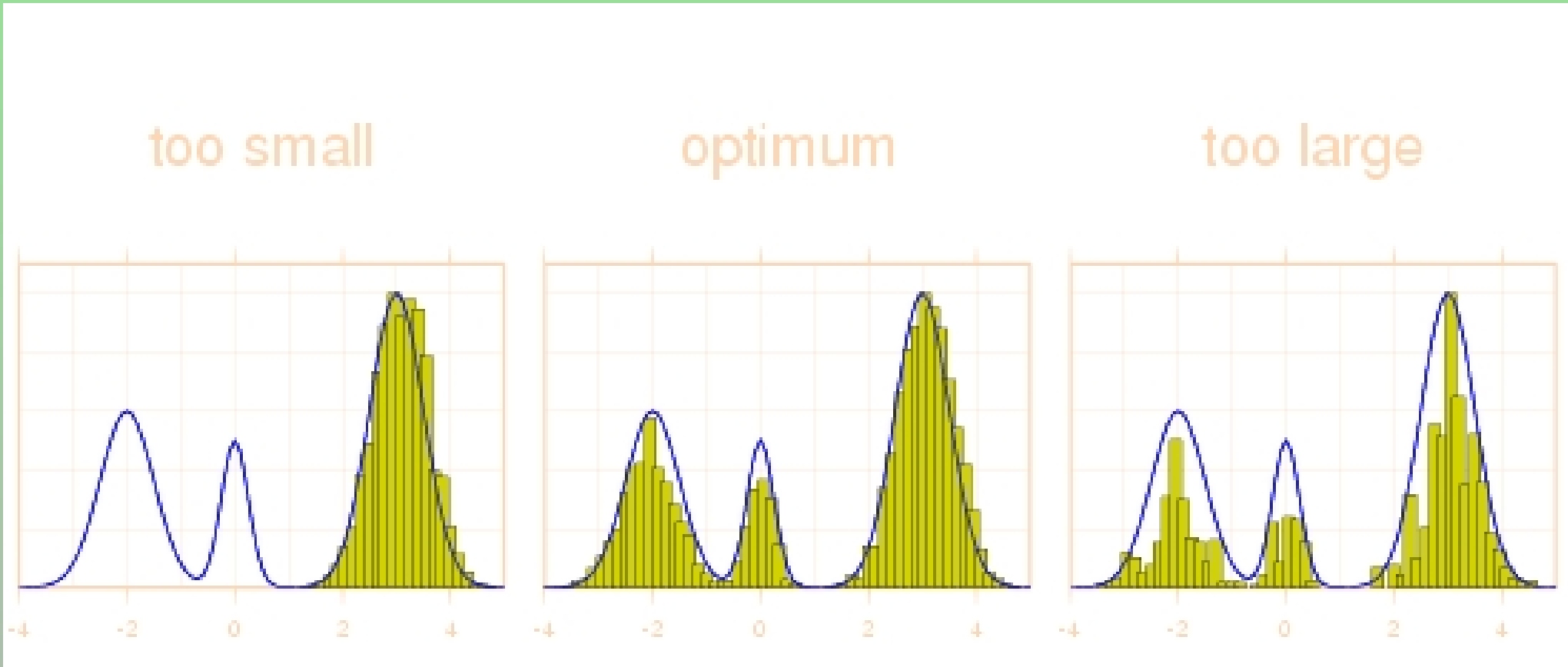
Algorytm Metropolisa-Hastinga

$$x_{i+1} = x_i + \delta x_i$$

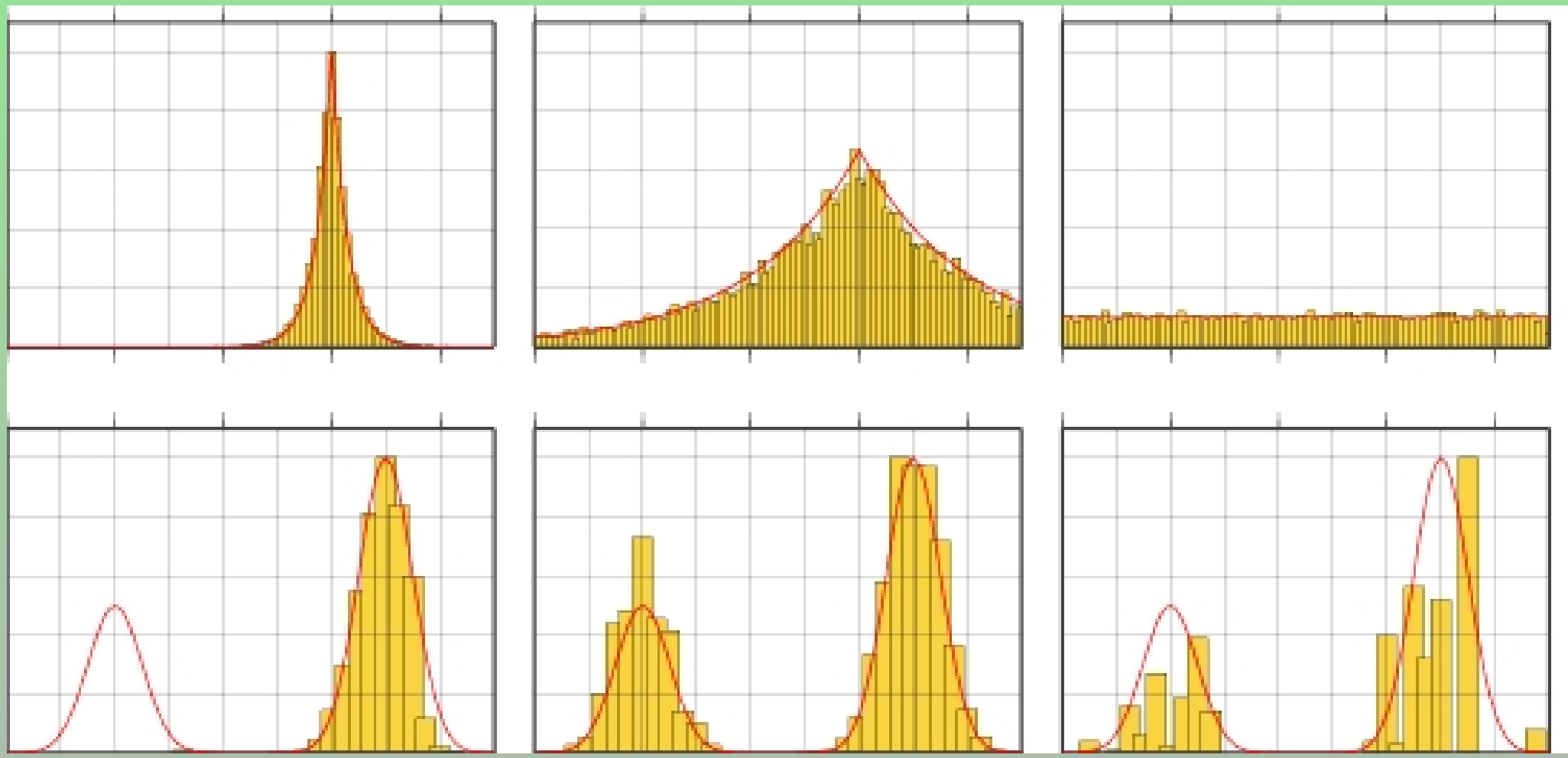
$$K(x_i; x_{i+1}) = \min\left\{1, \frac{p(x_{i+1})}{p(x_i)}\right\}$$

Algorytm MH - ograniczenia

$$x_{i+1} = x_i + \delta x$$



Algorytm MH - ograniczenia



- Utwórz rozkład "Boltzmanowski" $p(\mathbf{m}, T) = \exp(-S(\mathbf{m})/T)$
- Wybierz T_0, i model początkowy $\mathbf{m}^\alpha$ i oceń go

$$p_\alpha = p(\mathbf{m}^\alpha, T_0)$$

- Dla danej temperatury T_k wylosuj $\mathbf{m}^{\alpha+1}$ z $p(\mathbf{m}, T_k)$

  - ★ wylosuj próbke testową $\mathbf{m}^\beta$: $\mathbf{m}^\beta = \mathbf{m}^\alpha + \delta\mathbf{m}$
  - ★ oceń $\mathbf{m}^\beta$: $p_\beta = p(\mathbf{m}^\beta, T_k)$
  - ★ Utwóż $\mathbf{m}^{\alpha+1}$

    zaakceptuj $\mathbf{m}^\beta$ z prawdopodobińśtwem $p = min\,(1, p_\beta/p_\alpha)$

    $$\mathbf{m}^{\alpha+1} = \mathbf{m}^\beta$$

    jeśli $\mathbf{m}^\beta$ odrzucona duplikuj stan obecny

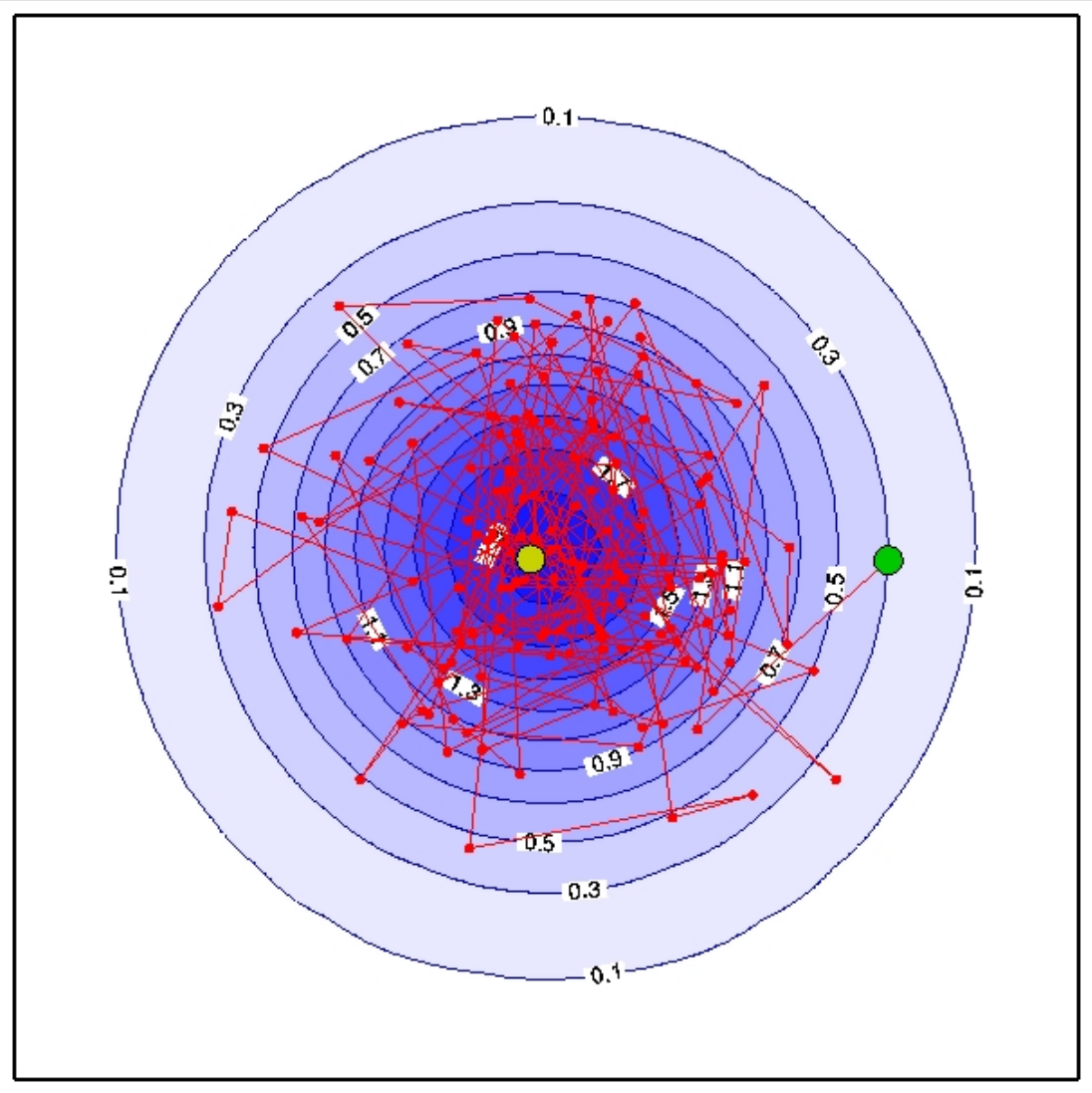
    $$\mathbf{m}^{\alpha+1} = \mathbf{m}^\alpha$$

- Zmniejsz temperaturę

$$T_{k+1} = f(T_k)$$

- Powtarzaj aż do osiągnięcia końcowej temperatury

Algorytm Metropolisa jako *Random Walk*



Algorytm SA przykład

```
while( T > Tf)    /* decrease T loop  */
  {
    nc =ceil( p2+p3*exp(-(double)k/(double)sa->n));
    if(nc<=1) nc=1;
    ++k;

   for(l=0;l<nc;l++) /* T=const loop */
    {

      for(j=0;j<N;j++)     /* generate new sample */
  {
  u=SArandval(0,1);        /* modified  VFSA */
    x= fabs(2*u -1);
  z= T*( pow(1+1/T,x) - 1);

  if(u >=0.5)
   p=mi[j] + z * (mu[j]-mi[j]);
  else
```

```
   p= mi[j] - z * (mi[j]-ml[j]);

  mf[j]=p;
  }

 tt= (*fun)(mf);   /* evaluate */
      e = tt-ta;
 z=exp(-e/T);
 u=SArandval(0,1);

 if( e<0 || z > u) /* accept  */
  {
    ta=tt;
   mc = mi;
   mi = mf;
   mf=mc;

if(ta < ts) /* keep the best */
 { ts=ta;
   for(i=0;i < N; i++)
            ms[i]=mi[i];
 }
```

```
 }

  } /* End of T=const loop */

  To=T;       /* decrease T (VFSA scheme) */
  z=Tc*pow((double)(k),di);
  T = Ti*exp(-z);

  if(sa->pf>0) /* messages to STD_ERR */
   { if(k%sa->pf==0)
      fprintf(stderr,"  VFSA:  T = %.2e   dT = %.2e   iter: %-6d  nc: %

   }

} /* End  of cooling (T) loop */
```