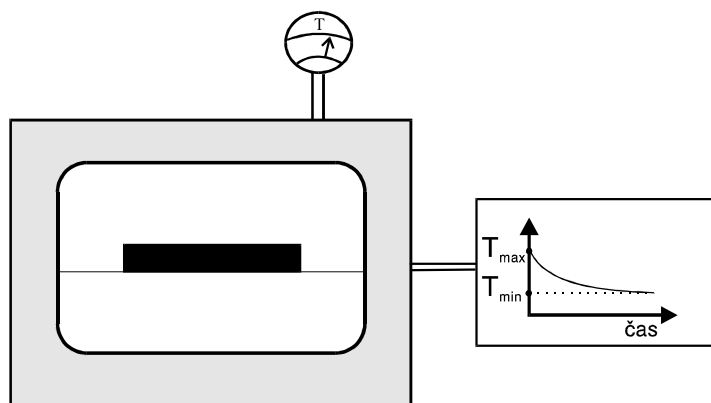


## 2. Simulované žihanie

### 2.1 Úvodné poznámky

Metóda simulovaného žihania [1,2] (Simulated Annealing - SA) patrí medzi tie stochastické optimalizačné algoritmy, ktoré, ako naznačuje už ich názov, majú základ vo fyzike (na rozdiel od iných stochastických optimalizačných algoritmov, ktoré majú svoj základ väčšinou v biológii). Algoritmus simulovaného žihania je založený na analógii medzi žiháním tuhých telies a optimalizačným problémom. Počiatkom 80-tych rokov Kirkpatrick, Gelatt a Vecchi [3] (Watson Research Center of the IBM, USA) a nezávisle Černý [4] (pôsobiaci na Katedre teoretickej fyziky MFF UK v Bratislave) dostali geniálny nápad, že problém hľadania globálneho minima sa môže byť realizovať podobným spôsobom ako žihanie tuhého telesa. Tento nový prístup k hľadaniu globálneho minima funkcie bol nazvaný *simulované žihanie*.



**Obrázok 2.1.** Znázornenie fyzikálnej realizácie žihania. Teleso sa vloží do pece (ľavý blok), ktorá je vyhriata na vysokú teplotu  $T_{max}$ . Teplota sa programovacím zariadením (pravý blok) pomaly znižuje na teplotu  $T_{min}$ . Týmto spôsobom sa odstránia štruktúrne defekty vyskytujúce sa v telese

Žihanie vo fyzike označuje taký proces, v ktorom je teleso umiestnené do pece vyhriatej na vysokú teplotu a postupným pomalým znižovaním teploty (pozri obr. 2.1) sa odstraňujú vnútorné defekty telesa. Pri vysokej teplote sú častice telesa (atómy alebo molekuly) náhodne usporiadané v priestore, takže teleso je roztopené. Potom sa teplota postupne znižuje, všetky častice telesa majú možnosť dostať sa do rovnovážnej polohy, čiže energia telesa sa znižuje. Budeme predpokladať, že proces ochladzovania je dostatočne pomalý, potom pre každú teplotu  $T$  je žihané teleso v tepelnej rovnováhe, ktorá je opísaná boltzmannovským rozdelením pravdepodobnosti, že pri teplote  $T$  je teleso v stave  $i$  s energiou  $E_i$

$$w_T(E_i) = \frac{1}{Q(T)} \exp\left(-\frac{E_i}{kT}\right) \quad (2.1a)$$

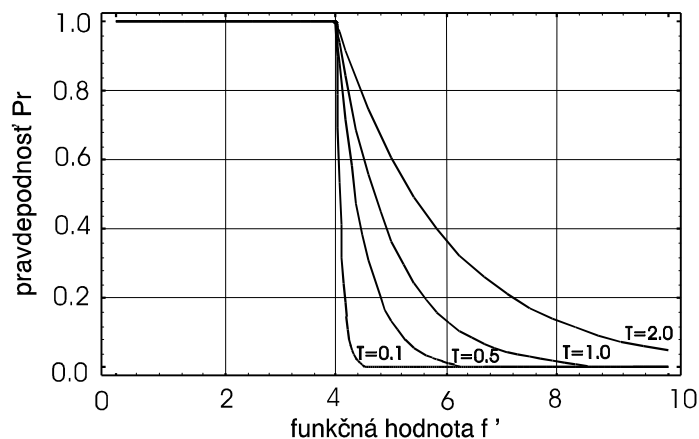
kde  $k$  je Boltzmannova konštanta a  $Q(T)$  je normalizačný faktor nazývaný *partičná funkcia*

$$Q(T) = \sum_i \exp\left(-\frac{E_i}{kT}\right) \quad (2.1b)$$

kde sčítanie obsahuje všetky stavy  $i$  telesa. Ak teplota  $T$  klesá, potom boltzmannovská distribúcia uprednostňuje stavy s menšou energiou. V prípade, že sa teplota blíži k nule, stav s minimálnou energiou má nenulovú (jednotkovú) pravdepodobnosť. Žiaľ, ako je dobre známe, ak je ochladzovanie telesa príliš rýchle (telesu nie je umožnené získať tepelnú rovnováhu pre každú teplotu  $T$ ), defekty v telese môžu "zamrznúť" za vzniku metastabilných štruktúr, ktoré sú vzdialené od mriežkovej štruktúry s najnižšou energiou.

Ako simulovať fyzikálnu evolúciu systému (napr. mnohočasticového kryštálu) smerom k tepelnej rovnováhe pre danú teplotu  $T$ ? Metropolis a spol. [5] navrhli metódu Monte Carlo, ktorá simuluje evolúciu systému tak, že generuje postupnosť stavov systému nasledujúcim spôsobom: Nech je daný aktuálny stav systému (určený polohou častíc telesa), potom sa malá náhodná porucha generuje tak, že častice sú "jemne" posunuté. Poznamenajme, že táto porucha musí byť "symetrická", t.j. pravdepodobnosť toho, že malou poruchou sa stav A zmení na stav B musí byť rovnaká ako pri zmene malou poruchou stavu B na stav A. Ak je rozdiel  $\Delta E = E_{\text{perturbed}} - E_{\text{current}}$  medzi porušeným stavom a aktuálnym stavom negatívny ( $E_{\text{perturbed}} < E_{\text{current}}$ ), potom proces pokračuje s novým porušeným stavom. V opačnom prípade, ak  $\Delta E \geq 0$ , pravdepodobnosť akceptovania porušeného stavu,  $Pr(\text{perturbed} \leftarrow \text{current})$  určuje exponenciála  $\exp(-\Delta E / kT)$  (pozri obr. 2.2)

$$Pr(\text{perturbed} \leftarrow \text{current}) = \min(1, \exp(-\Delta E / kT)) \quad (2.2)$$



**Obrázok 2.2.** Priebeh Metropolisovho kritéria  $Pr = \min(1, \exp(-(f'-f)/T))$  pre rôzne teploty  $T$ , kde  $f$  je fixná funkčná hodnota ( $f=4$ ) a  $f'$  je nezávislá premenná z intervalu  $[0,10]$ . Pre klesajúce hodnoty teploty  $T$  a pre  $f' > f$  pravdepodobnosť  $Pr \rightarrow 0$ , ak  $T \rightarrow 0$ .

Toto pravidlo akceptovania porušeného stavu sa nazýva *Metropolisovo kritérium*. Podľa tohto kritéria aplikovaním veľkého počtu porúch a ich akceptovaním do ďalšieho procesu s pravdepodobnosťou (2.2) dostaneme systém v tepelnej rovnováhe, pričom distribúcia pravdepodobnosti rozloženia stavov sa asymptoticky blíži k boltzmannovskej distribúcii (2.1). Tento tvar metódy Monte Carlo sa v štatistickej fyzike nazýva *Metropolisov algoritmus* [5]. K tomu, aby sme formalizovali Metropolisov algoritmus (pozri algoritmus 2.1) zavedieme nasledujúci formalizmus (ktorý je vhodný pre aplikáciu simulovaného žihania na minimalizáciu funkcií): stav systému nech je určený *stavovou premennou*  $\mathbf{x}$  (vo všeobecnosti vektor obsahujúci mnoho nezávislých reálnych premenných) a energiou  $f(\mathbf{x})$ . Proces porušenia stavu  $\mathbf{x}$  na iný stav  $\mathbf{x}'$  je formálne reprezentovaný stochastickým operátorom,

$\mathbf{x}' = O_{\text{pertur}}(\mathbf{x})$ . Stochastický charakter tohto operátora spočíva v náhodnej zmene stavu  $\mathbf{x}$  na stav  $\mathbf{x}'$ .

Ako už bolo poznamenané, Metropolisov algoritmus [5] produkuje distribúciu pravdepodobnosti stavov, ktorá sa asymptoticky blíži k Boltzmannovskej distribúcii (2.1)

$$w_T(\mathbf{x}) = \frac{1}{Q(T)} \exp\left(-\frac{f(\mathbf{x})}{T}\right) \quad (2.3a)$$

$$Q(T) = \sum_{\mathbf{x}} \exp\left(-\frac{f(\mathbf{x})}{T}\right) \quad (2.3b)$$

kde súčet obsahuje všetky stavy  $\mathbf{x}$ . Pre jednoduchosť sa Boltzmannova konštanta  $k$  teraz vynechala (pozri rovnice (2.1a-b), formálne bola táto konštanta zahrnutá do teploty  $T$ ).

```

procedure Metropolis_algorithm(input: $\mathbf{x}_{ini}$ ,  $k_{max}$ ,  $T$ ; output: $\mathbf{x}_{out}$ );
begin  $k := 0$ ;  $\mathbf{x} := \mathbf{x}_{ini}$ ;
    while  $k < k_{max}$  do
        begin  $k := k + 1$ ;
             $\mathbf{x}' := O_{\text{pertur}}(\mathbf{x})$ ;
             $Pr := \min(1, \exp(-(f(\mathbf{x}') - f(\mathbf{x}))/T))$ ;
            if random <  $Pr$  then  $\mathbf{x} := \mathbf{x}'$ ;
        end;
         $\mathbf{x}_{out} := \mathbf{x}$ ;
    end;

```

**Algoritmus 2.1.** Implementácia Metropolisovho algoritmu. Procedúra sa inicializuje tak, že počiatočný stav sa položí rovný vstupnému stavu  $\mathbf{x}_{ini}$ , opakuje sa  $k_{max}$ -krát (toto číslo musí byť dostatočne veľké, aby sa dosiahla tepelná rovnováha). Symbol  $O_{\text{pertur}}$  modifikuje aktuálny stav  $\mathbf{x}$  na  $\mathbf{x}'$ . Akceptovanie nového stavu sa rieši pomocou Metropolisovho kritéria realizovaného pre teplotu  $T$ . Po skončení Metropolisovho algoritmu výstupným stavom  $\mathbf{x}_{out}$  je posledný stav  $\mathbf{x}$ .

Metropolisov algoritmus sa môže použiť na počítačovú simuláciu žihania. V tomto prístupe sa simulované žihanie chápe ako postupnosť Metropolisových algoritmov realizovaných pre postupnosť vhodne sa znižujúcich teplôt, pričom výstupný stav  $\mathbf{x}_{out}$  z posledného Metropolisovho algoritmu slúži ako vstupný stav  $\mathbf{x}_{in}$  pre nasledujúci Metropolisov algoritmus. Táto procedúra sa inicializuje tak, že teplota  $T$  sa položí rovná maximálnej teplote  $T_{max}$  a Metropolisov algoritmus sa aplikuje tak dlho ( $k_{max}$ -krát, kde  $k_{max}$  je parameter Metropolisovho algoritmu), až sa dosiahne tepelná rovnováha. Potom sa teplota zníži (napr. pravidlom  $T := \alpha T$ , kde  $0 \ll \alpha < 1$ ) a znovu sa aplikuje Metropolisov algoritmus, pričom sa predpokladá, že sa dosiahne tepelná rovnováha. Algoritmus sa ukončí vtedy, keď teplota  $T$  dosiahne minimálnu hodnotu  $T_{min}$ , pre ktorú už žiadny stav s vyššou funkcionálnou hodnotou nie je akceptovaný (skoro nulová pravdepodobnosť akceptovania). "Zamrznutý" stav  $\mathbf{x}_{out}$  sa potom chápe ako výsledné riešenie (pozri algoritmus 2.2).

Jeden zo základných problémov simulovaného žihania je špecifikácia teplôt  $T_{max}$ ,  $T_{min}$  a pravidla pre znižovanie teploty  $T$ . V literatúre boli navrhnuté rôzne schémy, ako určiť tieto parametre simulovaného žihania [1,2,6]. Pretože náplňou tejto kapitoly nie je študovať podrobnú teóriu simulovaného žihania, budeme používať najjednoduchší prístup pre ochladzovanie (multiplikatívny prístup pomocou parametra  $\alpha$ ) a počiatočná teplota je zvolená tak, aby približne 50% stavov porušených stavov bolo akceptovaných Metropolisovým algoritmom.

```

procedure Simulated_annealing(input:Tmin,Tmax,kmax,α;output:xopt);
begin xini:=náhodne vygenerovaný stavový vektor;
      T:=Tmax;
      while T>Tmin do
        begin Metropolis_algorithm(xini,xout,kmax,T);
          xini:=xout;
          T:=α*T;
        end;
      xopt:=xout;
end;

```

**Algoritmus 2.2.** Implementácia simulovaného žihania, vstupné parametre sú  $T_{min}$ ,  $T_{max}$ ,  $k_{max}$ ,  $\alpha$ , výstupný parameter je  $\mathbf{x}_{opt}$ . Algoritmus sa inicializuje náhodným vygenerovaním počiatočného stavu  $\mathbf{x}_{ini}$  a maximálnou teplotou  $T_{max}$ . Cyklus **while** sa opakuje tak dlho, pokiaľ platí  $T > T_{min}$ , teplota  $T$  sa znižuje pomocou parametra  $\alpha$  pomocou vzťahu  $T:=\alpha*T$ . Po ukončení cyklu **while** sa výsledný stav  $\mathbf{x}_{out}$  považuje za výsledné riešenie označené  $\mathbf{x}_{opt}$ .

**Príklad 2.1.** Implementujte algoritmus simulovaného žihania (pozri algoritmus 2.2). Algoritmus špecifikujte pre funkciu  $n$  premenných, ktoré sú reprezentované binárne. Operátor mutácie  $O_{mut}$  je potom totožný s podobným operátorom z genetického algoritmu (pozri (3.2) a algoritmus 3.2).

Hlavný cieľ simulovaného žihania je nájsť globálne riešenie optimalizačného problému

$$\mathbf{x}_{opt} = \arg \min_{\mathbf{x} \in D} f(\mathbf{x}) \quad (2.4)$$

kde  $f(\mathbf{x})$  je reálna funkcia definovaná nad oblasťou  $D$  (ktorá je obvykle konečná a diskretná) a  $\mathbf{x}_{opt}$  je hodnota vektora nezávislých premenných, ktorá odpovedá globálnemu minimu funkcie  $f(\mathbf{x})$  nad doménou  $D$ . Premenná  $\mathbf{x}$  sa teraz chápe ako stavový vektor hypotetického fyzikálneho systému a hodnota funkcie  $f(\mathbf{x})$  vyjadruje jeho "energiu". Potom, vyššie uvedený optimalizačný problém (2.1) môže sa úspešne študovať metódou simulovaného žihania. Parameter "teplota" bude v tomto prístupe hrať úlohu radiaceho parametra metódy.

## 2.2 Vzťah k štatistickej fyzike

V predchádzajúcej časti tejto kapitoly bola naznačená blízka analógia medzi žiháním tuhého telesa (vo všeobecnosti fyzikálneho systému obsahujúceho mnoho častíc, ktoré medzi sebou interagujú) a riešením optimalizačného problému (2.4). Fyzikálny proces žihania sa môže úspešne modelovať na počítači pomocou Metropolisovho algoritmu. Tento algoritmus je založený na štatistickej fyzike, ktorá môže byť chápaná ako centrálna disciplína pre fyziku tuhej fázy. Ukážeme, že existuje veľmi tesný vzťah medzi optimalizačným problémom (2.4) a štatistickou fyzikou. Táto aplikácia simulovaného žihania nie je zaujímavá len ako určitá analógia, ale aj ako vhodný rámec pre štúdium konvergenčných vlastností simulovaného žihania.

Základný predpoklad štatistickej fyziky [7] je, že fyzika mnohočasticových systémov je kompatibilná so štatistickými súbormi a že časové stredné hodnoty mechanických veličín systému v rámci mikroskopickej rovnováhy sú rovné odpovedajúcim stredným hodnotám nad súbormi. V štatistickej fyzike sa definovali vhodné veličiny popisujúce fyzikálne vlastnosti systému, a tiež sa ukázalo, ako počítať tieto veličiny pomocou rovnovážneho rozdelenia stavov systému. Ukazuje sa, že v tepelnej rovnováhe je rozdelenie pravdepodobnosti určené boltzmannovským rozdelením (2.1a-b). Vzťah medzi štatistickou fyzikou a optimalizačným

problémom (2.4) sa môže teraz študovať vo veľmi špecifickej rovine. Predpokladajme, že máme hypotetický fyzikálny systém v tepelnej rovnováhe, ktorého vnútorné stavy  $\mathbf{x}$  (premenne z oblasti  $D$ ) majú distribúciu podľa vzťahu (2.3a-b) (formálne ekvivalentný s (2.1a-b)), potom pre tento optimalizačný problém môžeme definovať rôzne fyzikálne veličiny podobným spôsobom ako v štatistickej fyzike.

Uvažujem funkciu  $f$  definovanú nad diskretnou a konečnou oblasťou  $D$

$$f : D \rightarrow A \subset R_+ \quad (2.5)$$

kde  $A$  je oblasť (diskrétna a konečná podmnožina  $R_+$  obsahujúca nezáporné reálne čísla) funkčných hodnôt  $f$ . Rozdelenie pravdepodobnosti stavu  $\mathbf{x}$  (t.j. premennej  $\mathbf{x}$  z  $D$ ), ktoré je výsledkom Metropolisovho algoritmu realizovaného pre teplotu  $T$ , je určené vzťahmi (2.3a-b). Pre naše ďalšie úvahy bude vhodné poznať hustotu funkčných hodnôt  $y=f(\mathbf{x})$ , ktorá je určená takto

$$w_T(y) = \frac{|D(y)|}{Q(T)} \exp\left(-\frac{y}{T}\right) \quad (2.6a)$$

$$Q(T) = \sum_{y \in A} |D(y)| \exp\left(-\frac{y}{T}\right) \quad (2.6b)$$

kde  $D(y) \subset D$  je množina obsahujúca všetky  $x \in D$  pre ktoré platí  $y=f(\mathbf{x})$

$$D(y) = \{x \in D; y = f(x)\} \quad (2.6c)$$

a  $|D(y)|$  je jej kardinalita. Je jasné, že veličina  $Q(T)$  určená (2.6b) je identická s jej analógom (2.3b). Nasledujúce "makroskopické" veličiny sú definované takto

(1) *Stredná hodnota* funkcie  $f(\mathbf{x})$

$$\langle f \rangle_T = \sum_{y \in A} y w_T(y) \quad (2.7)$$

(2) *Stredná hodnota* funkcie  $f^2(\mathbf{x})$

$$\langle f^2 \rangle_T = \sum_{y \in A} y^2 w_T(y) \quad (2.8)$$

(3) *Disperzia* funkcie  $f(\mathbf{x})$

$$\sigma^2(T) = \sum_{y \in A} w_T(y) (\langle f \rangle_T - y)^2 = \langle f^2 \rangle_T - \langle f \rangle_T^2 \quad (2.9)$$

(3) *Entropia*

$$S(T) = - \sum_{y \in A} w_T(y) \ln \left( \frac{w_T(y)}{|D(y)|} \right) \quad (2.10)$$

(4) *Špecifické teplo*

$$C(T) = \frac{\partial}{\partial T} \langle f \rangle_T = \frac{\sigma^2(T)}{T^2} = T \frac{\partial S(T)}{\partial T} \quad (2.11)$$

Tento posledný vzťah sa môže jednoducho odvodiť derivovaním buď (2.7) alebo (2.10) podľa teploty  $T$ .

Hustota funkčných hodnôt (2.6a-b) vyhovuje týmto asymptotickým podmienkam

$$\lim_{T \rightarrow \infty} w_T(y) = w_\infty(y) = \frac{1}{|D|} |D(y)| \quad (2.12a)$$

$$\lim_{T \rightarrow 0} w_T(y) = w_0(y) = \delta(y, y_{opt}) = \begin{cases} 1 & (\text{pre } y = y_{opt}) \\ 0 & (\text{pre } y \neq y_{opt}) \end{cases} \quad (2.12b)$$

kde  $y_{opt} = f(x_{opt})$  je optimálna hodnota funkcie  $f(x)$  odpovedajúca bodu určenému vzťahom (2.4) ako globálne minimum. Druhý asymptotický vzťah (2.12b) je veľmi dôležitý pre metódu simulovaného žihania; ak sa podľa tohto vzťahu teplota  $T$  blíži k nule (cez rovnovážne stavy), potom *pravdepodobnosť výskytu stavu odpovedajúceho globálnemu minimu je jednotková*.

Parciálna derivácia  $\langle f \rangle_T$  podľa teploty  $T$  je určená vzťahom (2.11), potom

$$\frac{\partial \langle f \rangle_T}{\partial T} = \frac{\sigma^2(T)}{T^2} > 0 \quad (2.13)$$

To znamená, že stredná hodnota  $\langle f \rangle_T$  klesá, ak aj teplota  $T$  klesá, pomocou (2.12a-b) dostaneme

$$\lim_{T \rightarrow \infty} \langle f \rangle_T = \langle f \rangle_\infty = \frac{1}{|D|} \sum_{y \in A} |D(y)| y \quad (2.14a)$$

$$\lim_{T \rightarrow 0} \langle f \rangle_T = \langle f \rangle_0 = y_{opt} \quad (2.14b)$$

Stredná hodnota  $\langle f^2 \rangle_T$  vyhovuje podobným asymptotickým podmienkam

$$\lim_{T \rightarrow \infty} \langle f^2 \rangle_T = \langle f^2 \rangle_\infty = \frac{1}{|D|} \sum_{y \in A} |D(y)| y^2 \quad (2.15a)$$

$$\lim_{T \rightarrow 0} \langle f^2 \rangle_T = \langle f^2 \rangle_0 = y_{opt}^2 \quad (2.15b)$$

Pre disperziu (2.9) dostaneme

$$\lim_{T \rightarrow \infty} \sigma^2(T) = \sigma^2(\infty) = \langle f^2 \rangle_\infty - \langle f \rangle_\infty^2 \quad (2.16a)$$

$$\lim_{T \rightarrow 0} \sigma^2(T) = \sigma^2(0) = \langle f^2 \rangle_0 - \langle f \rangle_0^2 = y_{opt}^2 - (y_{opt})^2 = 0 \quad (2.16b)$$

Entropia (2.10) vyhovuje dvom dobre známym asymptotickým podmienkam

$$\lim_{T \rightarrow \infty} S(T) = S(\infty) = \ln |D| \quad (2.17a)$$

$$\begin{aligned} \lim_{T \rightarrow 0} S(T) &= S(0) = \ln |D(y_{opt})| \\ &= \ln 1 = 0 \quad (\text{pre } |D(y_{opt})| = 1) \end{aligned} \quad (2.17b)$$

Dôkaz týchto podmienok sa jednoducho uskutoční substitúciou (2.6) do (2.10) a potom asymptotickou zmenou teploty buď  $T \rightarrow \infty$  alebo  $T \rightarrow 0$ . Druhá vlastnosť (2.17b) je vo fyzike známa ako tretí zákon termodynamický. V prípade simulovaného žihania sa entropia môže byť interpretovať ako *kvantitatívna miera stupňa optimality*.

V dôsledku platnosti (2.11) sa parciálna derivácia entropie podľa teploty  $T$  určí takto

$$\frac{\partial S(T)}{\partial T} = \frac{\sigma^2(T)}{T^3} > 0 \quad (2.18)$$

Potom s klesajúcou teplotou  $T$  (cez rovnovážne stavy) musí klesať aj entropia  $S$ ,  $S(T) \rightarrow 0$  pre  $T \rightarrow 0$  (pozri rov. (2.17b)). V okolí bodu  $T=0$  aj entropia  $S(T)$  môže sa aproximovať mocninným rozvojom teploty  $T$

$$S(T) = A_0 + A_1 T + A_2 T^2 + \dots \quad (2.19)$$

kde  $A_0 = \ln(|D(y_{opt})|)$ , a  $A_1 > 0$ . Potom sa tepelná kapacita  $C(T)$  v okolí bodu  $T=0$  (pozri rov. (2.11)) aproximovaná takto

$$C(T) = T \frac{\partial S(T)}{\partial T} = T(A_1 + 2A_2 T + \dots) \quad (2.20)$$

V dôsledku platnosti tohto vzťahu a asymptotickej vlastnosti (2.16a) disperzie spĺňa tepelná kapacita  $C(T)$  tieto dve asymptotické podmienky

$$\lim_{T \rightarrow \infty} C(T) = \lim_{T \rightarrow 0} C(T) = 0 \quad (2.21)$$

To znamená, že  $C(T)$  musí mať pre  $0 < T < \infty$  aspoň jedno maximum a ak sa teplota asymptoticky blíži k nule alebo k nekonečnu, tepelná kapacita  $C(T)$  sa blíži k nule.

## 2.3 Modelové výpočty

Nech je zobrazenie  $f: D = \{0,1\}^N \rightarrow A \subset \mathbb{R}_+$ , s definičným oborom obsahujúcim binárne vektory dĺžky  $N$ , definované takto

$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_N) = \begin{cases} |\mathbf{x}| + 1 & (\text{pre } |\mathbf{x}| \leq p) \\ N - |\mathbf{x}| & (\text{pre } |\mathbf{x}| > p) \end{cases} \quad (2.22)$$

kde  $|\mathbf{x}| = \sum x_i$  je norma  $L_1$  vektora  $\mathbf{x} \in \{0,1\}^N$ ,  $0 \leq |\mathbf{x}| \leq N$ . Parameter  $p$  je ohraničený podmienkou  $0 \leq p \leq N$ . Pre  $p < N$  má táto funkcia dve minima, prvé minimum je  $f(\mathbf{x})=1$  pre  $|\mathbf{x}|=0$ , druhé (globálne) minimum je  $f(\mathbf{x})=0$  pre  $|\mathbf{x}|=N$ . Pre  $p=N$  má funkcia len jedno (t.j. globálne) minimum  $f(\mathbf{x})=1$  pre  $|\mathbf{x}|=0$ . Množina  $A$  funkčných hodnôt obsahuje nezáporné celé čísla, pričom  $A = \{0, 1, 2, \dots, f_{max}\}$ , kde  $f_{max} = \max(p+1, N-p-1)$ . Priebeh tejto funkcie je pre  $N=10$  a pre  $p=4$  a  $p=7$  znázornený na obrázku 2.3. Parameter  $p$  určuje stupeň *deceptívnosti* (falošnosti) tejto funkcie. Ak je  $p$  o trochu menšie ako  $N$ , potom funkcia obsahuje dlhé "úpätie" začínajúce v  $|\mathbf{x}|=0$  a končiacie v  $|\mathbf{x}|=p$ , za týmto úpäťím (pre  $p+1 \leq |\mathbf{x}| \leq N$ ) funkcia rýchle klesá k nule (pozri obr. 2.3, diagram B). Táto vlastnosť funkcie  $f(\mathbf{x})$  sa nazýva *deceptívnosť*, predstavuje ťažkú úlohu pre evolučné metódy, ak majú hľadať globálne minimum.

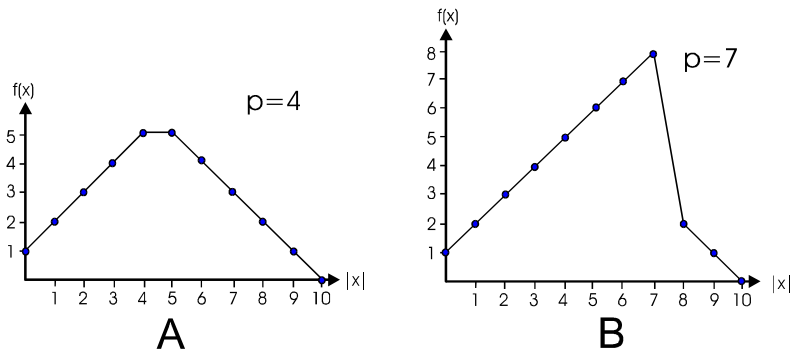
Operátor  $O_{perturb}$  z Metropolisovho algoritmu transformujúci binárny vektor  $\mathbf{x} \in \{0,1\}^N$  na iný binárny vektor  $\mathbf{x}' \in \{0,1\}^N$  sa v tomto prípade realizuje pomocou operátora mutácie  $O_{mut}^{(P_{mut})}$  (terminológia je prevzatá z genetického algoritmu)

$$\mathbf{x}' = (x'_1, x'_2, \dots, x'_N) = O_{mut}^{(P_{mut})}(\mathbf{x}) = O_{mut}^{(P_{mut})}(x_1, x_2, \dots, x_N) \quad (2.23)$$

kde sú jednotlivé zložky  $\mathbf{x}'$  sú určené takto

$$\forall i: x'_i = \begin{cases} 1 - x_i & (\text{ak } random < P_{mut}) \\ x_i & (\text{ostatné prípady}) \end{cases} \quad (2.24)$$

kde  $P_{mut}$  je pravdepodobnosť "mutácie" premennej  $x_i$  (pre  $\forall i$ ) na jej komplement (ak  $random < P_{mut}$ ) alebo zostane nezmenená (ostatné prípady) a  $random$  je náhodné číslo z intervalu  $[0,1)$  s rovnomerným rozdelením pravdepodobnosti. Pravdepodobnosť  $P_{mut}$  je veľmi malá ( $0 < P_{mut} \ll 1$ ), v opačnom prípade by mutačný operátor produkoval stavy, ktoré sú veľmi vzdialené od pôvodného stavu  $\mathbf{x}$ .



**Obrázok 2.3.** Priebeh funkcie (2.22) pre  $N=10$  a  $p=4, 7$ . Pre  $p=7$  funkcia obsahuje "dlhé" úpätie za ktorým je oblasť prudkého spádu do globálneho minima.

Aby sme mohli vypočítať makroskopické veličiny definované v predchádzajúcej kapitole 2.2 musíme poznať kardinality množín  $D(y)$  pre funkciu (2.19). Tak napr. pre  $N=10$  a pre  $p=4$  a  $p=7$  sú tieto kardinality určené takto

$$\begin{aligned}
 p=4 : & |D(0)|=1, \quad |D(1)|=11, \quad |D(2)|=55, \quad |D(3)|=165, \quad |D(4)|=330, \quad |D(5)|=462 \\
 p=7 : & |D(0)|=1, \quad |D(1)|=11, \quad |D(2)|=55, \quad |D(3)|=45, \quad |D(4)|=120, \quad |D(5)|=210, \\
 & |D(6)|=252, \quad |D(7)|=210, \quad |D(8)|=120
 \end{aligned}$$

Priebehy makroskopických veličín pre funkciu  $f(\mathbf{x})$  špecifikovanú konštantami  $N=10$  a  $p=1,4,7,9$  sú znázornené na obr. 2.4, diagramy A - E. Ako z tohto obrázka vyplýva, všetky asymptotické vlastnosti sú splnené, ktoré boli predpovedané v predchádzajúcej časti tejto kapitoly pre makroskopické veličiny. Predovšetkým stredná hodnota  $\langle f \rangle_{T \rightarrow 0} \rightarrow 0$  pre  $T \rightarrow 0$ . To znamená, že pre malé teploty je dominantný s jednotkovou pravdepodobnosťou výskytu stav s nulovou funkčnou hodnotou.

Pre simulované žihanie realizované na počítači, výpočet makroskopických veličín môže byť uskutočnený pomocou približných hustôt funkčných hodnôt, ktoré sú aproximované výskytom funkčných hodnôt akceptovanými Metropolisovým algoritmom pre fixnú teplotu  $T$ . Nech  $\chi_T(y)$  je počet výskytov stavu  $x$ , ktorý odpovedá funkčnej hodnote  $y$  a ktoré sa vyskytujú v Metropolisovom algoritme pre danú teplotu  $T$ . Potom, hustota  $w_T(y)$  môže byť približne určená takto

$$w_T(y) \approx \frac{\chi_T(y)}{\sum_{y' \in A} \chi_T(y')} \quad (2.25)$$

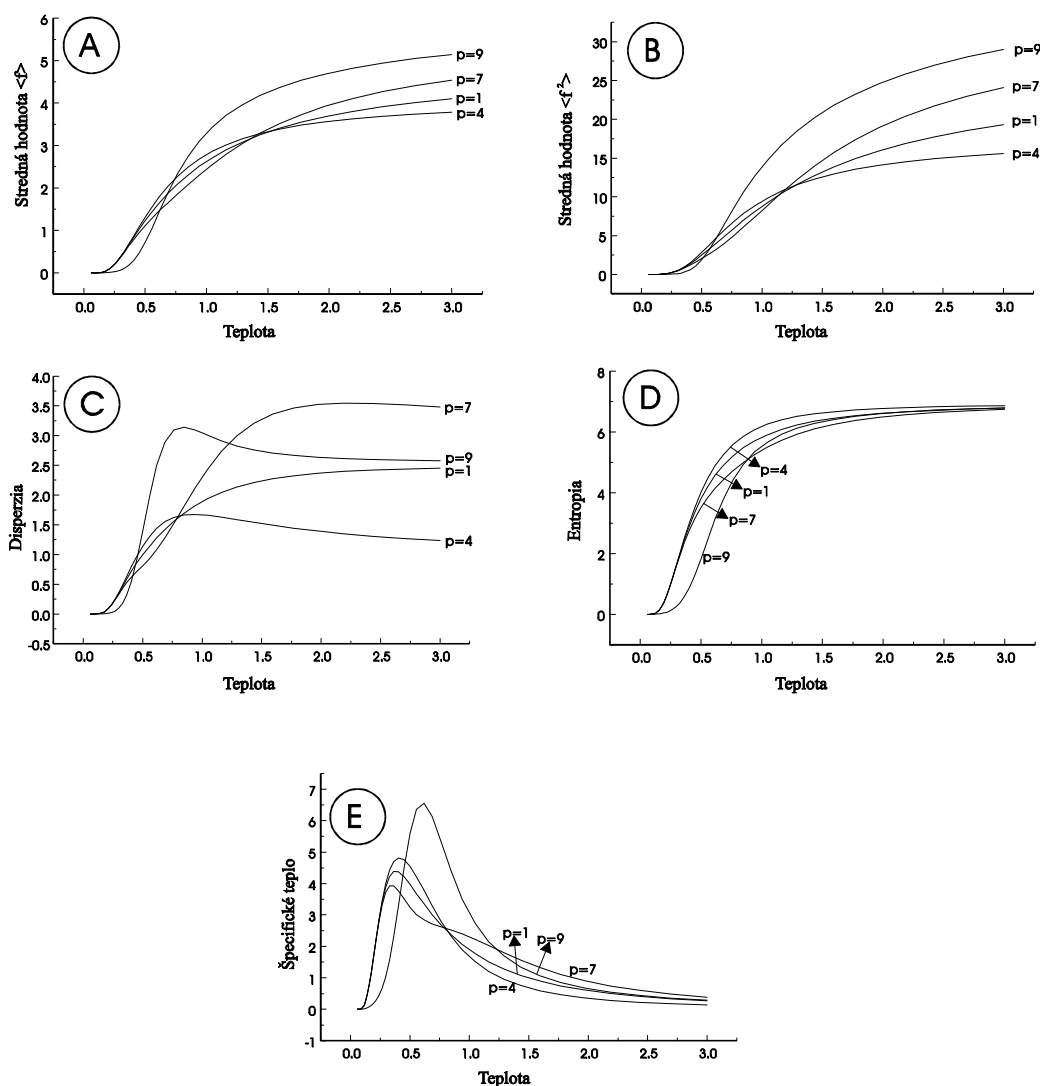
Pre  $k_{max}$  dostatočne veľké, táto aproximácia by mala poskytovať výsledky, ktoré sú veľmi blízke k presným hodnotám  $w_T(y)$ . Priebehy makroskopických vlastností (2.7-11) s hustotami aproximovanými vzťahom (2.25) sú znázornené na obr. 2.5, diagramy A - E. Parametre simulovaného žihania použité vo výpočtoch mali tieto hodnoty

$$k_{max}=10^4, \quad P_{mut}=0.1, \quad T_{max}=3, \quad T_{min}=3/50, \quad \text{and } \alpha=0.95 \quad (2.26)$$

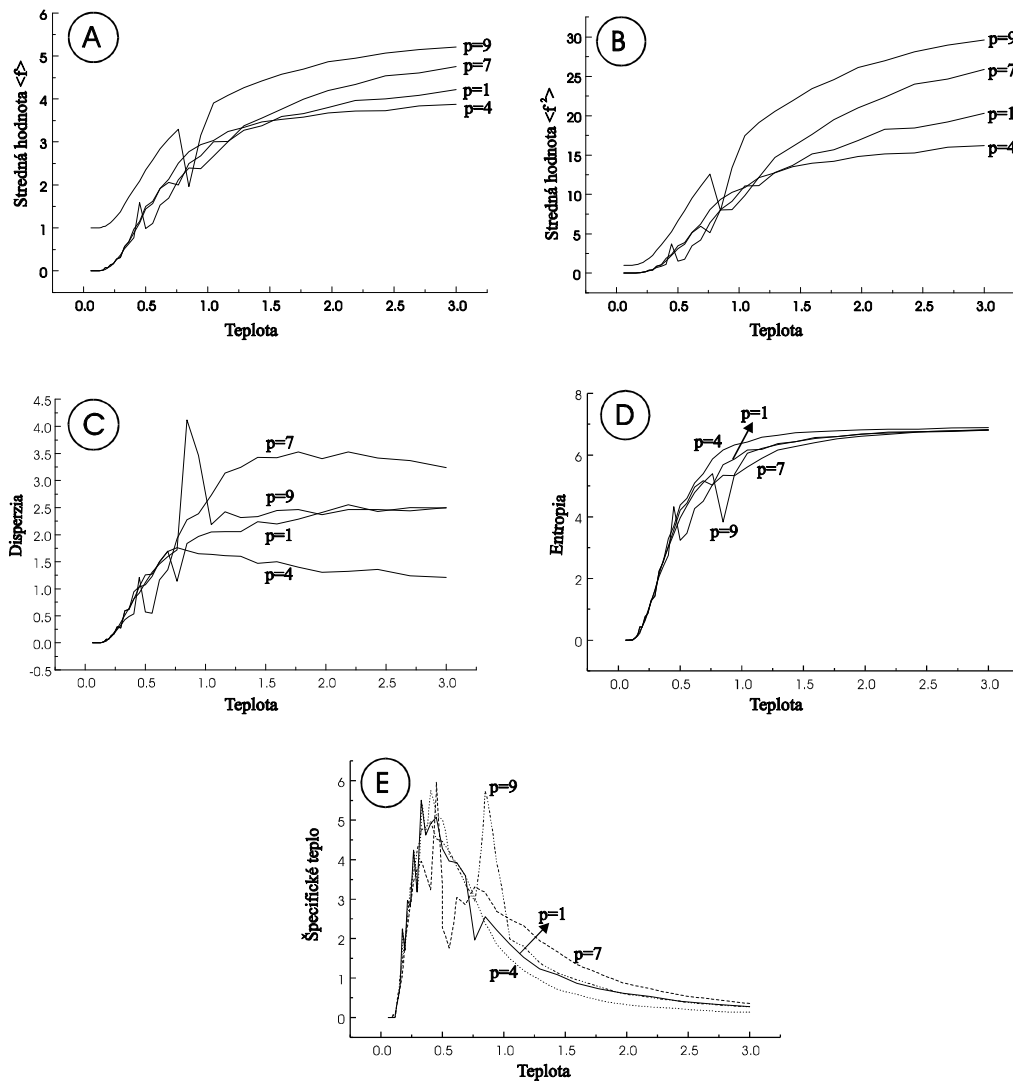
Vidíme, že simulované žihanie pre  $p=1,4$  a  $7$  poskytuje správne výsledky pre strednú hodnotu  $\langle f \rangle_T$  (t.j.  $\langle f \rangle_{T \rightarrow 0} \rightarrow 0$  pre  $T \rightarrow 0$ ), ale pre  $p=9$  (funkcia  $f(\mathbf{x})$  má najväčší stupeň deceptívnosti) metóda simulovaného žihania nie je schopná poskytnúť korektný výsledok (dostali sme  $\langle f \rangle_{T \rightarrow 0} \rightarrow 1$  pre  $T \rightarrow 0$ ). To znamená, že pre  $p=9$  je simulované žihanie schopné nájsť len suboptimálne riešenie  $x=(0,0,\dots,0)$ ,  $|x|=0$ , pre ktoré platí  $f(x)=1$ . Existuje interval kritických teplôt  $[0.7,0.9]$ , nad ktorým metóda poskytuje hustoty stavu  $y=0$ , ktoré sú blízke



teoretickým hodnotám, avšak vo vnútri intervalu výskyt stavov s  $y=0$  bol skoro nulový. To znamená, že pre nižšie teploty nie je Metropolisov algoritmus nie vytvárať stavy s  $y=0$  (v dôsledku širokej bariéry funkcionálnych hodnôt, pozri obr. 2.3, diagram B). Túto skutočnosť indikujú napr. vysoké maximá pre disperziu a špecifické teplo, ktoré sa môžu fyzikálne interpretovať ako "fázové prechody" medzi konfiguráciami, ktoré sú separované vysokými bariérami funkčných hodnôt. To znamená, že tieto dve makroskopické veličiny sa môžu byť použiť ako vhodný indikátor tých tepelných intervalov, kde znižovanie teploty  $T$  musí byť veľmi pomalé, aby sa úspešne prekonal vyššie spomínaný "fázový prechod".



**Obrázok 2.4.** Priebehy makroskopických veličín vypočítaných teoreticky pomocou vzorcov (2.7-11) pre parametre  $\rho=1,4,7, 9$  (diagramy A - E).



**Obrázok 2.5.** Priebeh makroskopických veličín vypočítaných metódou simulovaného žihanie, ktorá je špecifikovaná parametrami (2.26). Pre prípad s najväčšou deceptívnosťou ( $p=9$ ), odpovedá finálny stav  $\mathbf{x}_{opt}$  jednotkovej funkčnej hodnote (lokálne minimum), t.j. metóda nebola schopná poskytnúť globálne minimum.

Uvedieme niekoľko vysvetľujúcich poznámok k simulovanému žihaniu aplikovanému na deceptívnu funkciu  $f(\mathbf{x})$  s hodnotou parametra  $p=9$ . Ukázali sme, že simulované žihanie s parametrami (2.26) nie je schopné nájsť správne riešenie. Vo všeobecnosti sa táto neschopnosť môže jednoducho prekonať tak, že pravdepodobnosť  $P_{mut}$  sa podstatne zvýši, napr.  $P_{mut}=0.5$ . Avšak potom simulované žihanie sa nemôže chápať ako evolučná optimalizačná metóda, prehľadávanie priestoru riešení je viac náhodné ako systematické. Simulované žihanie je pre vysoké hodnoty pravdepodobnosti  $P_{mut}$  redukované na "slepé" hľadanie. Pre malé dimenzie môže byť tento prístup úspešný, pre veľké dimenzie je tento prístup úplne beznádejný. V nasledujúcej časti tejto kapitoly uvedieme také dve modifikácie

štandardnej verzii simulovaného žihania, že budú schopné poskytnúť správne riešenie aj pre vysoko deceptívne funkcie a súčasne bude zachovaný evolučný charakter metódy.

### 2.3.1 Simulované žihanie s elitizmom

V originálnej verzii simulovaného žihania slúži výsledný stav z aplikácie Metropolisovho algoritmu (realizovaného pri teplote  $T$ ) ako počiatočný stav nasledujúceho Metropolisovho algoritmu (realizovaného pri zníženej teplote  $\alpha T$ ). Tento základný predpoklad simulovaného žihania bude teraz modifikovaný tak, že Metropolisov algoritmus sa inicializuje najlepším riešením, ktoré sa získalo v priebehu predchádzajúcej histórie metódy (t.j. pre epochy simulovaného žihania s vyššími teplotami ako súčasná teplota); takúto jednoduchú modifikáciu simulovaného žihania budeme nazývať *simulované žihanie s elitizmom* [8]. Modifikácia procedúr `Metropolis_algorithm` a `Simulated_annealing`, ktoré boli uvedené v predchádzajúcej časti tejto kapitoly sa realizuje algoritmami 2.3 a 2.4.

```

procedure Elitism_Metropolis_algorithm(input : $x_{in}$ ,  $k_{max}$ ,  $T$ ;
                                         output: $x_{opt}$ );

begin  $k:=0$ ;  $x:=x_{in}$ ;  $x_{opt}:=x_{in}$ ;
      while  $k<k_{max}$  do
        begin  $k:=k+1$ ;
           $x':=O_{pertur}(x)$ ;
           $Pr:=\min(1, \exp(-(f(x')-f(x))/T))$ ;
          if  $\text{random}<Pr$  then
            begin  $x:=x'$ ;
              if  $f(x')<f(x_{opt})$  then  $x_{opt}:=x'$ ;
            end;
          end;
        end;
      end;

```

**Algoritmus 2.3.** Implementácia Metropolisovho algoritmu so zahrnutým elitizmom. Vstupné parametre sú  $x_{in}$ ,  $k_{max}$ ,  $T$ . Algoritmus je inicializovaný vstupným vektorom  $x_{in}$ , najlepšie riešenie je uložené do  $x_{opt}$ , výsledná hodnota tohto vektora slúži ako výstupný parameter procedúry.

```

procedure Elitism_Simulated_annealing(input : $T_{min}$ ,  $T_{max}$ ,  $k_{max}$ ,  $\alpha$ ;
                                         output: $x_{opt}$ );

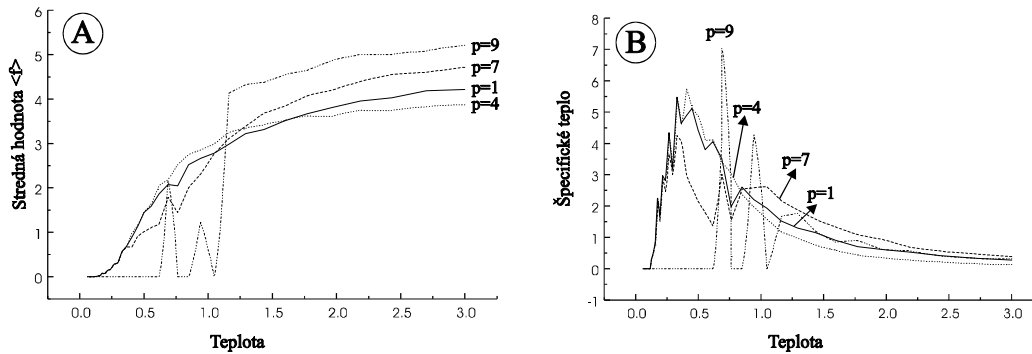
begin  $x_{opt}:=$ náhodne vygerovaný stavový vektor;
       $T:=T_{max}$ ;
      while  $T>T_{min}$  do
        begin Forced_Metropolis_algorithm( $x_{opt}$ ,  $k_{max}$ ;  $T$ );
           $T:=\alpha*T$ ;
        end;
      end;

```

**Algoritmus 2.4.** Implementácia simulovaného žihania s elitizmom. Vstupné parametre do procedúry sú  $T_{min}$ ,  $T_{max}$ ,  $k_{max}$ ,  $\alpha$  a výstupný parameter je  $x_{opt}$ . Náhodne inicializovaný stav  $x_{opt}$ , zaznamenáva najlepšie riešenie vyskytujúce sa v celej histórii algoritmu.

Vo všeobecnosti nespĺňa simulované žihanie s elitizmom dôležitú podmienku metódy, že konštanta  $k_{max}$  je dostatočne veľká a že pre diskrétny kroky teploty je Metropolisov algoritmus v tepelnej rovnováhe (t.j. stavy majú distribúciu podľa Boltzmannu (2.3)). Pretože v každom kroku zníženia teploty sa Metropolisov algoritmus inicializuje najlepším riešením,

ktoré bolo zaznamenané v celej predchádzajúcej histórii algoritmu, môže nastať taká situácia (predovšetkým pre nižšie teploty), že výsledné riešenie produkované posledným behom Metropolisovho algoritmu odpovedá lokálnemu minimu, ktoré je podstatne odlišné od globálneho minima.



**Obrázok 2.2.** Priebeh strednej hodnoty  $\langle f \rangle$  a špecifického tepla, ktoré sa spočítali pomocou simulovaného žihania s elitizmom. Pre všetky hodnoty parametra  $p$  výsledný stav  $x_{opt}$  odpovedá globálnemu minimu s nulovou funkčnou hodnotou (pripomeňme, že štandardná verzia simulovaného žihania poskytuje pre  $p=9$  len lokálne minimum s jednotkovou funkčnou hodnotou).

Numerické výsledky získané simulovaným žiháním s elitizmom pre funkciu (2.22) špecifikovanú hodnotami  $N=10$  a  $p=1, 4, 7, 9$  pre strednú hodnotu  $\langle f \rangle$  a špecifickú teplotu, sú znázornené na obr. 2.6, diagramy A a B. Vidíme, že pre  $p=9$  (najdeceptívnejší tvar funkcie) sú zmeny na priebehu veľmi dramatické. Pre stredné teploty nebol Metropolisov algoritmus schopný udržať nenulovú distribúciu stavu s nulovou funkčnou hodnotou. Pretože najlepšie riešenie s nulovou funkčnou hodnotou sa zaznamenalo už v prvých etapách simulovaného žihania, Metropolisov algoritmus produkuje pre malé teploty ( $T < 0.6$ ) akceptované stavy s nulovou funkčnou hodnotou, ktoré odpovedajú globálnemu minimu.

Slabšia verzia simulovaného žihania s elitizmom je založená na požiadavke, aby sa Metropolisov algoritmus inicializoval najlepším riešením získaným len v predchádzajúcom kroku, a nie v celej predchádzajúcej histórii algoritmu. Žiaľ, tento prístup k najdeceptívnejšej forme optimalizovanej funkcie pre  $p=9$  nebol úspešný pri prekonávaní veľkej bariéry so spádom do lokálneho minima (s jednotkovou funkčnou hodnotou)

### 2.3.2 Paralelné simulované žihanie

Táto verzia simulovaného žihania sa s úspechom použila na riešenie komplikovaných kombinatoriálnych grafovo-teoretických problémov [9,10], kde štandardná verzia simulovaného žihania nie je schopná poskytnúť správne globálne riešenie. Základná idea paralelného simulovaného žihania spočíva v súčasnom aplikovaní simulovaného žihania na množinu  $P$  stavov  $x, x', x'', \dots$  ktoré sú "synchronizované" tak, že Metropolisove algoritmy bežiacie nad nimi majú rovnakú teplotu. Aby stavy medzi sebou interagovali, zaviedla sa operácia *kríženia* medzi stavmi podobným spôsobom ako v genetických algoritmoch. Nech  $x, x' \in P$  sú dva náhodné vybrané stavy, operácia kríženia vytvorí z týchto dvoch "rodičovských" stavov dva nové stavy - potomkov

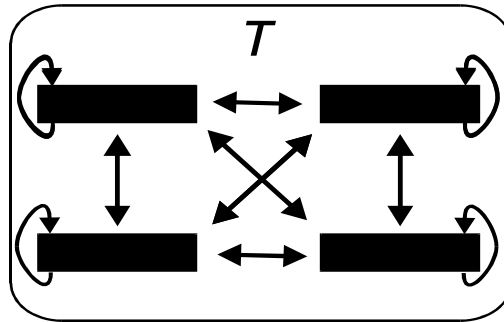
$$(z, z') = O_{cross}(x, x') \quad (2.27)$$

Formálne, nech  $x=(x_1,x_2,\dots,x_N)$  and  $x'=(x'_1,x'_2,\dots,x'_N)$  sú dva rodičovské stavy vyjadrené ako reťazce znakov dĺžky  $N$ , potom potomkovia sú definovaní takto

$$z=(x_1,\dots,x_r,x'_{r+1},\dots,x'_N) \quad (2.28a)$$

$$z'=(x'_1,\dots,x'_r,x_{r+1},\dots,x_N) \quad (2.28b)$$

kde  $r$  je náhodne vybraný bod križenia,  $1 < r < N$ .



**Obrázok 2.7.** Schematické znázornenie paralelného simulovaného žihania. Na jednotlivé stavy (znázornené čiernymi blokmi) sa aplikovali nezávislé Metropolisove algoritmy s rovnakou teplotou  $T$ . Sípky znázorňujú, že stavy medzi sebou interagujú s malou pravdepodobnosťou  $P_{cross}$ .

```

procedure Parallel_Metropolis_algorithm(input:  $P_{ini}, k_{max}, T$ ;
                                         output:  $P_{out}$ );
begin  $k:=0$ ;  $P:=P_{ini}$ ;
      while  $k > k_{max}$  do
        begin  $k:=k+1$ ;
          if random  $< P_{cross}$  then
            begin vyber náhodne  $x \in P$ ;
               $x' := O_{pertur}(x)$ ;
               $Pr := \min(1, \exp(-(f(x') - f(x))/T))$ ;
              if random  $< Pr$  then  $P := (P \setminus \{x\}) \cup \{x'\}$ ;
            end else
              begin vyber náhodne  $x, x' \in P$ ;
                 $(z, z') := O_{cross}(x, x')$ ;
                 $Pr_1 := \min(1, \exp(-(f(z) - f(x))/T))$ ;
                 $Pr_2 := \min(1, \exp(-(f(z') - f(x'))/T))$ ;
                if random  $< Pr_1$  then  $P := (P \setminus \{x\}) \cup \{z\}$ ;
                if random  $< Pr_2$  then  $P := (P \setminus \{x'\}) \cup \{z'\}$ ;
              end;
            end;
           $P_{out} := P$ ;
        end;
end;

```

**Algoritmus 2.5.** Implementácia paralelného Metropolisovho algoritmu. Vstupné parametre sú  $P_{ini}$  (počiatočná množina stavov),  $k_{max}$ ,  $T$ , výstupný parameter je  $P_{out}$  (konečná množina stavov). Vonkajší cyklus **while** sa opakuje  $k_{max}$ -krát. Pravdepodobnosť  $P_{cross}$  určuje náhodný výber jednoduchej poruchy vybraného stavu  $x$ , alebo či sa vykoná križenie nad dvojicou náhodne vybraných stavov  $x$  a  $x'$ . Dvojica nových stavov, ktoré sú produkované križením, je akceptovaná do množiny stavov pomocou Metropolisovho kritéria aplikovaného nezávisle nad dvoma dvojicami stavov  $(x, z)$  a  $(x', z')$ .

Proces interakcie dvoch náhodne vybraných stavov  $x, x' \in P$  je v rámci Metropolisovho algoritmu určený pravdepodobnosťou  $0 < P_{cross} \ll 1$ . Táto pravdepodobnosť určuje proces výmeny, t.j. namiesto mutácie stavu  $x \in P$  sa uskutoční interakcia dvoch stavov  $x, x' \in P$  pomocou operátora kríženia (2.27). Potom sa otázka, či sú nové stavy  $z$  a  $z'$  akceptované, sa rieši pomocou Metropolisovho kritéria (2.2) aplikovaného zvlášť pre páry  $(x, z)$  a  $(x', z')$ . Napríklad, ak Metropolisovo kritérium akceptuje nový stav  $z$ , potom sa množina  $P$  obnovuje tak, že starý stav  $x$  sa nahradí novým stavom  $z$ , formálne  $P := (P \setminus \{x\}) \cup \{z\}$ . Výsledné riešenie sa určí ako najlepšie riešenie z množiny  $P$

$$x_{opt} = \arg \min_{x \in P} f(x) \quad (2.29)$$

kde  $P$  je množina riešení rezultujúca z dokončeného simulovaného žihania. Procedúry `Metropolis_algorithm` a `Simulated_annealing` sa môžu jednoducho modifikovať pre potreby paralelného simulovaného žihania (pozri algoritmy 2.5 a 2.6).

```

procedure Parallel_Simulated_annealing(input:  $T_{max}, T_{min}, k_{max}, \alpha$ ;
                                         output:  $x_{opt}$ );
begin P:=náhodne generovaný súbor stavov  $x, x', \dots$ ;
      T:= $T_{max}$ ;
      while  $T > T_{min}$  do
        begin  $P_{ini} := P$ ;
              Parallel_Metropolis_algorithm( $P_{ini}, k_{max}, T, P_{out}$ );
               $P := P_{out}$ ;
               $T := \alpha * T$ ;
        end;
       $x_{opt} := \arg \min \{f(x), x \in P\}$ ;
end;

```

**Algoritmus 2.2.** Implementácia paralelného simulovaného žihania. Vstupné parametre sú  $T_{max}$ ,  $T_{min}$ ,  $k_{max}$ ,  $\alpha$ , výstupný parameter je  $x_{opt}$ . Algoritmus sa inicializuje náhodnou generáciou množiny stavov z  $P$ . Po skončení algoritmu je najlepší stav z  $P$  označený  $x_{opt}$ .

**Príklad 2.2.** Formuláciou simulovaného žihania sme zavřšili celú plejádu evolučných optimalizačných algoritmov: (1) horolezecký algoritmus (rôzne verzie, s učením, metóda zakázaného prehľadávania, evolučné programovanie), (2) genetické algoritmy a (3) simulované žihanie. Zvoľte si určité modelové úlohy a na týchto testujte efektívnosť vyššie uvedených metód a porovnajte ich.

## Literatúra

- [1] P.J.M. van Laarhoven and E.H.L. Aarts: *Simulated Annealing. Theory and Applications*. Reidel, Dordrecht 1987.
- [2] R.H.J.M. Otten and L.P.P.P. van Ginneken: *Annealing Algorithm*. Kluwer, Boston 1989.
- [3] S. Kirkpatrick, C.D. Gelatt Jr. and M.P. Vecchi: Optimization by Simulated Annealing. *Science* **220** (1983) 671-680.
- [4] J. Černý: Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *J. Opt. Theory Appl.* **45** (1985) 41-51.

- [5] N. Metropolis, A.W. Rosenbluth, N.N. Rosenbluth, A.H. Teller and E. Teller: Equation of State Calculations for Fast Computing Machines. *J. Chem. Phys.* **21** (1953) 1087-1092.
- [6] A L. Ingber: Very fast simulated re-annealing. *J. Mathematical Computer Modelling*, **12** (8) (1989) 967-973, available with code at <http://www.ingber.com/>
- [7] M. Toda, R. Kubo, and N. Saito: *Statistical Physics*. Springer Verlag, Berlin 1983.
- [8] V. Kvasnička and J. Pospíchal: Simulated Annealing. *MATCH (Communications in Mathematical and Computer Chemistry)* **34** (1996) 7-49.
- [9] S.W. Mahfoud and D.E. Goldberg: A genetic algorithm for parallel simulated annealing. In R. Manner and B. Manderick (eds.): *Parallel Problem Solving from Nature*, 2, pp. 301-310, North-Holland, Amsterdam 1992.
- [10] J. Pospíchal and V. Kvasnička: Fast Evaluation of Chemical Distance by Simulated Annealing Algorithm. *J. Chem. Inf. Comp. Sci.* **33** (1993) 879-885.