

# 26 ZUM – Princip metod hill-climbing a tabu search. Simulované žíhaní.

Matej Choma

June 3, 2019

V tomto handoute sa zameriame na algoritmy iteratívnej optimalizácie. Tieto metódy riešia obecný optimalizačný problém (s ľubovoľne komplexnou štruktúrou) spôsobom “pokus-omyl”. Znalosti o štruktúre problému typicky získavajú až počas behu.

**Definícia 1 (Optimalizačný problém)** *Nech  $X$  je množina všetkých prípustných riešení a zobrazenie  $f : X \rightarrow \mathbb{R}$  je **kriteriálna funkcia**, ktorá hodnotí kvalitu nášho riešenia. Optimalizačný problém je formulovaný ako hľadanie  $\mathbf{x}^* \in X$  maximalizujúceho funkčnú hodnotu  $f(\mathbf{x})$ :*

$$\mathbf{x}^* \in \arg \max_{\mathbf{x} \in X} f(\mathbf{x}).$$

## 1 Hill climbing

Metóda **hill climbing** generuje v každom kroku riešenie blízke ku doteraz najlepšiemu riešeniu (obecne susedné stavy zo stavového priestoru). Pokiaľ nájdeme lepšie riešenie, nahradíme to najlepšie a pokračujeme. Je tu analógia s kráčaním do kopca – pred každým krokom sa rozhliadneme a vykročíme hore.

**Steepest hill climbing** je obmenou algoritmu, v ktorej vygenerujeme viacero nových riešení a z nich vyberieme to najlepšie.

Metódy hill climbingu trpia na nasledovné problémy:

- pokiaľ sa dostaneme na lokálny extrém, nemáme ako pokračovať, pretože všetky riešenia v okolí sú horšie
- s rastúcou dimenziou rastie objem priestoru exponenciálne a prestáva byť možné dostatočne navzorkovať podobné riešenia v okolí

## 2 Simulované žíhaní

Simulované žíhaní rozšíri hill climbing o parameter  $t$  (analógia – teplota pri žíhaní ocele, pri zahriatí a pozvoľnom ochladzovaní sa atómy pekne ukladajú do kryštalickej mriežky), ktorý sa s časom znižuje. Pokiaľ neexistuje v okolí riešenia  $x$  lepšie riešenie, preskočíme na horšie riešenie  $y$  podľa pravdepodobnosti závislej na:

- miere zhoršenia  $(f(x) - f(y)) \rightarrow$  čím väčšie zhoršenie, tým menšia pravdepodobnosť
- aktuálnej teplote  $t \rightarrow$  čím väčšia teplota, tým väčšia pravdepodobnosť

## 3 Tabu search

Snaží sa zabrániť oscilácií medzi riešeniami a tak prinútiť optimalizačný algoritmus, aby sa vymanil z lokálneho optima. Tabu search zavádza **tabu list**, v ktorom sa ukladajú všetky už nájdené riešenia. Optimalizačný algoritmus sa nemôže nikdy vrátiť ku riešeniu, ktoré už raz objavil. Po objavení lokálneho optima je teda algoritmus nútený prejsť ku horším riešeniam a tak získať potenciál sa opäť zlepšiť.