



Heuristické optimalizačné procesy



Lokálne prehľadávanie

Marian.Mach@tuke.sk

<http://neuron.tuke.sk/~machm>

Február, 2013

Lokálne prehľadávanie

- Cieľom je hľadať riešenie v rámci celého priestoru kandidátov
 - globálne riešenie
opt: $x \in S \mid f(x) < f(y), \text{ kde } y \in S \wedge x \neq y$
rozh: $x \in S \mid \text{valid}(x)$
- sústredenie sa iba na malý podpriestor
- okolie - lokálny pohľad:
 - lokálne riešenie s ohľadom na okolie
opt: $\text{lok}(x) \neq \text{glob}(x)$
rozh: $\text{lok}(x) = \text{glob}(x)$

Okolie

- Okolie N je časť priestoru kandidátov S
 - zahŕňa validných aj nevalidných kandidátov
 - podmnožina kandidátov, ktorí sú „blízko“ nejakému kandidátovi
 - vždy voči nejakému kandidátovi $\rightarrow N(s), s \in S$
- Dva spôsoby definovania
 - na základe vzdialenosti
 - na základe vymenovania
 - explicitné vymenovanie
 - pomocou mapovania (procedúrou/vlastnosťou)

Okolie - vzdialenosť

- Existencia funkcie vzdialenosti

$$\text{dist}: S \times S \rightarrow R$$

- Okolie je dané ako

$$N(x) = \{ y \in S \mid \text{dist}(x,y) \leq \varepsilon \} \quad \text{pre } 0 \leq \varepsilon, x \in S$$

- Príklady (štandardné okolia)

- $\text{dist}(x,y) = (\sum (|x_i - y_i|)^n)^{1/n}$

- Euklidova vzdialenosť (n=2)

- Hammingova vzdialenosť (n=1)

Okolie - mapovanie

- Existencia mapovacej funkcie

$$\text{map}: S \rightarrow 2^S$$

- Okolie $N(x)$ je dané ako zoznam tých kandidátov, ktorí sú mapovaní ako súčasť okolia kandidáta x

- Príklady (štandardné okolia)

- k-exchange okolie

- 1-flip (SAT)
- 2-swap (TSP)

Graf okolia

- Relácia okolia indukuje orientovaný graf na priestore kandidátov
- Vlastnosti
 - ak relácia okolia je symetrická, tak graf je neorientovaný
 - stupeň vrcholu = veľkosť okolia
 - regulárnosť grafu
 - priemer grafu

Komponenty lokálneho prehľadávania

▪ Algoritmus lokálneho prehľadávania pre inštanciu π vyžaduje tieto komponenty:

▪ $S(\pi), N(\pi), M(\pi)$

▪ inicializačná funkcia

$$init(\pi): \square \rightarrow D(S(\pi) \times M(\pi))$$

▪ kroková funkcia

$$step(\pi): S(\pi) \times M(\pi) \rightarrow D(S(\pi) \times M(\pi))$$

▪ ukončovací predikát

$$term(\pi): S(\pi) \times M(\pi) \rightarrow D(\{\top, \perp\})$$

Štruktúra LS – rozhodovací problém

- input: π
- output: $r \in S \mid \square$
- $(s, m) = \text{init}()$
- **while**(not $\text{term}(s, m)$)
- $(s, m) = \text{step}(s, m)$
- **endwhile**
- **if**($\text{valid}(s)$) **then**
- **return** s
- **else**
- **return** \square
- **endif**

Štruktúra LS – optimalizačný problém

- input: π
- output: $r \in S \mid \square$
- $(s, m) = \text{init}()$
- $r = s$
- **while**(not $\text{term}(s, m)$)
- $(s, m) = \text{step}(s, m)$
- **if**($f(s) < f(r)$) **then**
- $r = s$
- **endif**
- **endwhile**
- **if**($\text{valid}(r)$) **then**
- **return** r
- **else**
- **return** \square
- **endif**

Prehľadávacia trajektória

- Závisí na použitých funkciách *init()* a *step()*
- Krok hľadania: $(s_1, s_2) \in S \times S$
 - *step*: $s_2 = \text{step}(s_1)$
- Trajektória:
 - $(s_0, s_1, \dots, s_n) \mid s_0 \in S, (s_i, s_{i+1}) \in S \times S$
 - *step*(... (*step*(*init*()))))
- Príklady
 - URP (uninformed random picking)
 - URW (uninformed random walk)