Heuristické optimalizačné procesy

Lokálne prehľadávanie, iteračné vylepšovanie

prednáška 4

Ing. Ján Magyar, PhD.

ak. rok. 2024/2025 ZS

Lokálne prehľadávanie

cieľom je hľadať riešenie v rámci celého priestoru kandidátov

globálne riešenie

- optimalizácia: $x \in S \mid f(x) < f(y)$, kde $y \in S \wedge x \neq y$
- rozhodovanie: $x \in S \mid valid(x)$

sústreďuje sa iba na malý podpriestor

Lokálne riešenie s ohľadom na okolie

optimalizácia: lok(x) $\neq$ glob(x)

rozhodovanie: lok(x) = glob(x)

Okolie

okolie N je časť priestoru kandidátov S ($N \subset S$)

- zahŕňa validných aj nevalidných kandidátov
- podmnožina kandidátov "blízkych" istému kandidátovi: $N(s)$, $s \in S$

definovanie okolia

- na základe vzdialenosti
- na základe vymenovania
    - explicitné (pre diskrétny priestor)
    - pomocou mapovania (procedúrou alebo vlastnosťou)

Okolie - vzdialenosť

existencia funkcie vzdialenosti $dist\colon S \times S \rightarrow R$

okolie je dané ako $N(x) = \{y \in S \mid dist(x, y) \leq \varepsilon\}$ *pre* $0 \leq \varepsilon,\ x \in S$

štandardné okolia: $dist(x, y) = (\Sigma\, (|\, x_i - y_i\, |)^n\,)^{1/n}$

- Euklidova vzdialenosť ($n = 2$)
- Hammingova vzdialenosť ($n = 1$)

Okolie - mapovanie

existencia mapovej funkcie *map*: $S \rightarrow 2^S$

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

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
- dostupnosť grafu
- priemer grafu

Komponenty lokálneho prehľadávania

algoritmus lokálneho prehľadávania pre inštanciu π vyžaduje 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 $term(\pi): S(\pi) \times M(\pi) \rightarrow D(\, \{\top, \bot\} \,)$

Štruktúra LS - rozhodovací problém

input: π
output: $s \in S$ □

(s, m) = *init*(π)
while(**not** *term*(s, m))
    (s, m) = *step*(s, m)
endwhile
if(*valid*(s)) **then**
    return s
else
    return □
endif

Štruktúra LS - optimalizačný problém

input: π
output: $r \in S$

```
( s, m ) = init(π)
r = s
while( not term( s, m ) )
        ( s, m ) = step( s, m )
        if( f( s ) < f( r )  ) then
                r = s
        endif
endwhile
if( valid(r) ) then
        return r
else
        return
endif
```

Prehľadávacia trajektória

závisí od použitých funkcií *init()* a *step()*

krok hľadania: $(s_1, s_2) \in S \times S$, kde $s_2 = step(s_1)$

trajektória: $(s_0, s_1, \ldots, s_n) \mid s_0 \in S, (s_i, s_{i+1}) \in S \times S$, teda
$step(\ldots (step(init())))$

príklady

- URP (uninformed random picking)
- URW (uninformed random walk)

Iteračné vylepšovanie

Informovaná stratégia

neinformovanou stratégiou nedokážeme budovať na predchádzajúcich skúsenostiach a tak smerovať k riešeniu

ohodnocovacia funkcia $g: S \rightarrow R$

- ohodnocuje kvalitu kandidátov v aktuálnom okolí
- jej globálne minimum súhlasí s optimom úlohy

rozhodovací a optimalizačný problém možno riešiť rovnakým spôsobom

Výber ohodnocovacej funkcie

problémovo závislý ale aj nezávislý

rozhodovací problém
- optimalizačná podoba úlohy
- štandardná podoba

optimalizačný problém
- použitie cieľovej funkcie
- použitie inej funkcie so zhodnou polohou globálneho optima

Príklad: SAT, MAXSAT

Delta (inkrementálna) evaluácia

Iteračné vylepšovanie

Funkcia *step*

$I(s) = \{ x \in N(s) \mid g(x) > g(s) \}$

step: $p(x) = 1 / \#I(s)$ ak $x \in I(s)$

$p(x) = 0$ inak

maximalizácia (existuje aj minimalizačná podoba)

Funkcia *init* (URP)

init: $p(x) = 1 / \#S$ pre všetky $x \in S$

Predikát *term*

$\#I(s) = 0$ dosiahnutie lokálneho optima

Algoritmus

input: π
output: $s \in S$□

s = *urp*()
while(#I(s) > 0)
    s = *select*(I(s))
endwhile
if(*valid*(s)) **then**
    return s
else
    return□
endif

Uviaznutie v lokálnom optime

lokálne optimum - algoritmus nevie urobiť žiadny krok

ak lokálne optimum = globálne optimum, tak našli sme optimálne riešenie

ak lokálne optimum ≠ globálne optimum, tak

- našli sme suboptimálne riešenie
- jeho kvalita sa nedá teoreticky vopred odhadnúť
- algoritmus uviazol v lokálnom optime

otázky?