

PSO – Optimalizace rojem částic (Particle Swarm Optimization)

Algoritmy PSO patří k metodám prohledávání prostoru. Jsou příbuzné s genetickými algoritmy.

PSO, optimalizace rojem částic, vychází z existence určité populace nějakých částic, které tvoří roj směřující ke globálnímu optimu. Inspirace pochází například od hejna ryb, které se drží pohromadě a směřují k nalezení co nejvíce co nejlepší potravu. Roj či hejno se pohybuje určitým nenáhodným směrem, který je částečně narušován nepředvídatelnými odbočováním do stran. Směr pohybu je v každém časovém kroku dán v principu dvěma složkami:

- momentálním globálním optimem a
- optimem momentálně nejlepší z částic.

Podobnost s genetickými algoritmy (GA) spočívá ve vytvoření počáteční populace náhodných řešení, optimum se hledá aktualizací generace. Na rozdíl od GA však PSO nepoužívá evoluční operátory typu křížení, mutace, apod. Částice, představující potenciální řešení, se pohybují prohledávaným prostorem vždy směrem udávaným momentálně nejlepší částicí z roje (rybou z hejna). PSO je jednoduché na implementaci, nemá mnoho parametrů, je náročné na výkon paměti i procesoru. PSO lze používat tam, kde GA, a naopak, i když samozřejmě výsledky nemusejí být totožné a zaručené. V principu jde o stochastickou optimalizační techniku, vycházející z odvětví *artificial life* (umělý život). Tyto techniky původně byly vyvíjeny pro simulaci a studium živých organismů a jevů s nimi spojených, ale obecně je lze aplikovat na nejrůznější výpočetní problémy.

PSO je zaměřeno na “společenské” biologické systémy, kde se zkoumá chování jedinců tvořících součásti kolektivu, jejich vzájemné interakce a interakce s okolím (existuje název *swarm intelligence*, tj. inteligence roje/hejna/davu). Používá se například ke zkoumání nepředvídatelné dynamiky společenského chování různých živých organismů (ptáci, ryby, mravenci, atd.).

Nyní existují dvě hlavní skupiny: optimalizace mravenčí kolonie (souboru mravenců), ACO (*ant colony optimization*) a PSO (*particle swarm optimization*). Používají se pro diskrétní optimalizační problémy.

Princip PSO inspirovaný hejnem ptáků

Skupina ptáků náhodně hledá v určité oblasti potravu, a je v ní pouze jeden kus potravy. O potravě vědí jen někteří ptáci, ne všichni. Jako nejlepší strategie se používá následování ptáka, jemuž je ta potrava nejbliž. Každá částice v roji představuje jednoho ptáka v prohledávaném prostoru a má určitou hodnotu své kvality, kterou stanovuje funkce přizpůsobenosti (*fitness function*). Kromě toho má každá částice nějakou rychlost, s níž se pohybuje a která udává i směr pohybu. Částice se pohybují prostorem tak, že následují tu, která je momentálně nejlepší. Hledání optima se provádí iterativně. V každém iteračním kroku je každá částice aktualizována pomocí dvou “optimálních” hodnot:

- doposud nalezeným nejlepším řešením každé částice p_{best} (*particle best*), a
- doposud nejlepším řešením populace g_{best} (*global best*).

Po nalezení obou nejlepších hodnot p_{best} a g_{best} jsou částice upravovány z hlediska jejich rychlosti a polohy pomocí dvou jednoduchých vztahů:

$$(1) \quad \mathbf{v} \leftarrow \mathbf{v} + c_1 \cdot \text{rand}() \cdot (p_{best} - \text{present}) + c_2 \cdot \text{rand}() \cdot (g_{best} - \text{present}), \text{ a}$$
$$(2) \quad \text{present} \leftarrow \text{present} + \mathbf{v},$$

kde \mathbf{v} je rychlost částice, present je její současná kvalita, a $\text{rand}()$ je náhodné číslo $\in [0.0, 1.0]$. Konstanty c_1 a c_2 jsou tzv. akcelerační konstanty, ovlivňující míru aktualizace částic, např. 0.5 nebo 2.0 nebo 1.0, apod.

Pseudokód algoritmu

```
do
  for each particle
    spočítej hodnotu přizpůsobenosti  $f$ 
    if( $f$ ) >  $p_{best}$  nastav současnou hodnotu jako novou  $p_{best}$ 
  end

  vyber hodnotu částice s nejlepší přizpůsobeností
  jako  $g_{best}$ 

  for each particle
    spočítej rychlost částice dle vztahu (1)
    aktualizuj pozici částice dle vztahu (2)
  end
while není dosažen max. počet iteračních kroků nebo
min. hodnota chyby
```

Rychlosti částic (složky vektoru rychlosti) jsou v každé dimenzi omezeny na uživatelem stanovenou hodnotu v_{max} , která se použije při překročení limitu rychlosti.

Použití PSO

Pro aplikaci PSO je nutno 1) vhodně representovat řešený problém a 2) navrhnout funkci přizpůsobenosti (*fitness*). PSO obecně pracuje s reálnými čísly.

Parametry PSO:

- počet částic – řádově desítky, obvykle 20-40 (málo částic snižuje výpočetní složitost, ale zároveň i možnost nalést optimum; hodně částic vede k opačným výhodám a nevýhodám – někdy se používají stovky částic);
- počet dimenzí částic – je dán optimalizovaným problémem;
- rozsah hodnot částic – je dán optimalizovaným problémem;
- v_{max} – ovlivňuje maximální možnou změnu rychlosti částice při jedné iteraci;
- konstanty učení c_1 a c_2 – obvykle mívají hodnotu 2.0, experimenty používají nejčastěji rozsah hodnot [0, 4];
- podmínka ukončení – minimální chyba požadovaná uživatelem a předem zadaná (funkce přizpůsobenosti se použije pro výpočet chyby v každé iteraci), nebo stanovení max. počtu iteračních kroků (stovky, tisíce).

Obvykle je nutno vhodné parametry a jejich kombinaci najít experimentálně.