**Dokumentácia**

**Cat Swarm Optimization**

**Lukáš Gereg**

**Martin Matta**

**Barbora Kubalcová**

**Martina Marasová**

**Technická univerzita v Košiciach Katedra kybernetiky a umelej inteligencie**

**Fakulta elektrotechniky a informatiky Inteligentné systémy, 2. ročník**

Obsah

[Mačací algoritmus (CSO) – popis a inšpirácia 3](#_Toc121509996)

[ Postup hlavnej časti algoritmu 4](#_Toc121509997)

[ Matematický model 4](#_Toc121509998)

[o Hľadací mód (Seeking mode) 4](#_Toc121509999)

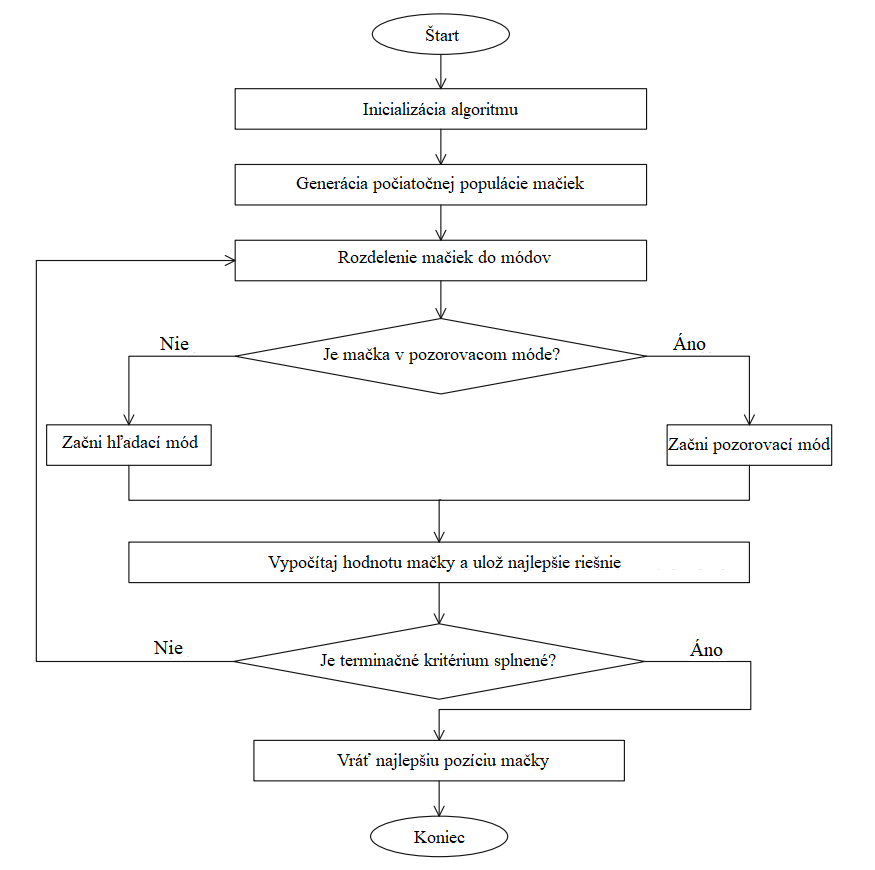
[o Prenasledovací mód (tracing mode) 5](#_Toc121510000)

# Mačací algoritmus (CSO) – popis a inšpirácia

CSO je algoritmus inšpirovaný správaním mačiek v reálnom svete. **Prírodné mačky** strávia väčšinu času v pokoji, sústredia sa na pozorovanie svojho okolia a striehnu na korisť. Ak zaznamenajú nejaký pohyb, pustia sa do akcie a začnú loviť. **Umelá mačka** má preto dva módy – pozorovací a sledovací.

* **Hľadací mód** (seeking mode) - mačka stojí na mieste a pozoruje okolie
* **Prenasledovací mód** (tracking mode) - mačka prenasleduje korisť a mení svoju pozíciu

Na začiatku algoritmu vytvoríme populáciu mačiek a náhodne ich rozmiestnime po   
m-dimenzionálnom priestore. Každá mačka v tomto priestore predstavuje jedno riešenie. Mačky následne rozdelíme do dvoch skupín – hľadací a sledovací mód. Ako sa už spomínalo, väčšinu svojho času strávia pozorovaním, preto bude skupina pozorujúcich mačiek väčšia. Pomer mačiek v skupinách nám udáva **MR** (mixture ratio). Po rozdelení vypočítame hodnotu mačky (potencionálneho riešenie) pomocou ohodnocovacej funkcie. Najlepšie nájdené riešenie sa uloží do pamäte.



Obr. 1 – Vývojový diagram CSO algoritmu

## Postup hlavnej časti algoritmu

1. Vytvor inicializačnú populáciu mačiek, umiestni ich náhodne do *m*-dimenzionálneho priestoru a každej mačke priraď náhodnú rýchlosť.
2. Každej mačke priraď mód vo zvolenom pomere (MR) a pridaj ju ku skupine mačiek s tým istým módom.
3. Pre každú mačku vypočítaj hodnotu a do pamäte ulož najlepšiu mačku. Najlepšie riešenie predstavuje pozícia mačky.
4. Na základe prideleného módu, vykonaj hľadanie alebo pozorovanie.
5. Ak je splnené terminačné kritérium, ukonči proces, inak sa vráť ku kroku 2.

## Matematický model

Každá mačka je umiestnená v *m*-dimenzionálnom priestore kandidátov, a teda má *m* súradníc, ktoré určujú jej pozíciu *Xn*, a nejakú určitú rýchlosť *vn*. Ďalej má označenie, v ktorom z módov sa aktuálne nachádza a hodnotu z ohodnocovacej funkcie.

### Hľadací mód (Seeking mode)

V tomto móde mačka ostáva na mieste a pozoruje svoje okolie. V prípade detekcie koristi (poprípade nebezpečenstva) sa rozhodne zmeniť svoju polohu na základe štyroch faktorov:

* SMP (seeking memory pool) – počet vytvorených kópií (klonov) mačky
* SRD (seeking range of the selected dimension) – „ako ďaleko“ sa vie mačka pozrieť (posun v dimenzii)
* CDC (counts of dimension to change) – koľko dimenzií bude zmenených
* SPC (self-position consideration) – boolovská hodnota. Rozhodnutie mačky ostať na aktuálnej pozícii alebo odísť

#### Postup

1. Urob SMP počet kópií každej mačkyi. Ak je SPC pravdivé, vytvorí sa (SMP - 1) kópií mačky a jedna ostane na pôvodnom mieste.
2. Na základe CDC vypočítaj novú pozíciu pre každú kópiu mačky pomocou rovnice

kde

*Xcn* je nová pozícia mačky,

*Xc* je aktuálna pozícia,

*R* je náhodné číslo z intervalu <0, 1>

1. Vypočítaj hodnoty mačiek pre ich nové pozície. Ak sú všetky hodnoty z ohodnocovacej funkcie rovnaké, nastav pravdepodobnosť výberu na 1 pre všetkých kandidátov, inak vypočítaj pravdepodobnosť výberu podľa vzorca

, kde 0 < *i < j*

kde

*Pi* je pravdepodobnosť výberu aktuálneho kandidáta

*FSi* hodnota mačky z ohodnocovacej funkcie

*FSmax* maximálna hodnota ohodnocovacej funkcie

*FSmin*  minimálna hodnota ohodnocovacej funkcie

Riešime

*FSb = FSmax* pre minimalizačný problém

*FSb = FSmin* pre maximalizačný problém

1. Pomocou vybranej stratégie (roul-wheel, tournament, best fitness-self, random) náhodne vyber cieľový bod a zmeň pozíciu mačkyi.

### Prenasledovací mód (tracing mode)