

Partikelschwarmoptimierung in Python

Martin Cresnoverh

25.02.2009

Ziele:

- Globale Optimierung mit einem Partikelschwarm
- Programmierung in Python
- Visualisierung mit Mayavi2

Vorteile eines Partikelschwarms bei globalen Optimierungsproblemen:

- Durch “Kommunikation” zwischen den Partikeln ist die Gefahr, in ein lokales Extremum zu fallen, gering.
- Kann einfach mit anderen Algorithmen erweitert werden.
- Relativ schnell.

Aufbau

- Initialisierung der Schwarmpartikel auf zufälligen Positionen innerhalb des Suchraumes.
- Iteration bis Abbruchkriterium erreicht:
 - Evaluierung der Fitness der Partikel
 - Bestimmung der nächsten Geschwindigkeit
 - Bewegung der Partikel an die neuen Positionen

Schwarmpartikel “wissen” die Position des

- besten von allen Schwärmen bisher erreichten Wertes. (*gbest*)
- besten persönlich erreichten Wertes. (*pbest*)
- besten in der Nachbarschaft erreichten Wertes. (*nbest*)

Die Geschwindigkeit v jedes Partikels errechnet sich für jeden Iterationsschritt i aus

$$v_i = v_{i-1} \cdot w_k + o_1 \cdot \text{rand}() \cdot (pbest - x_i) \\ + o_2 \cdot \text{rand}() \cdot (gbest - x_i) + o_3 \cdot \text{rand}() \cdot (nbest - x_i)$$

$o_1 \dots o_3$... Optimierungsparameter, normalerweise $o_1 = o_2 = o_3$

$\text{rand}()$... Gleichverteilte Zufallszahl zwischen 0 und 1

x_i ... Position des Partikels im Iterationsschritt i

w_k ... Trägheit des Partikels

Parameter

- Anzahl der Partikel
- Anzahl der Iterationen bzw. Abbruchkriterium
- Anzahl der Nachbarschaftspartikel
- Maximale Geschwindigkeit der Partikel
- Optimierungsparameter $\sigma_1 \dots \sigma_3$ zur Gewichtung der einzelnen Bestwerte (persönlich, Nachbarschaft, global)
- Trägheit der Partikel

Programmierung in Python mit der Erweiterung NumPy

- bietet Matlab-ähnliche Arrays
- Matlab-ähnliche Performance
- Opensource
- sehr portabel (Linux/Unix, Windows, Symbian...)

Visualisierung mit Mayavi2

- basiert auf VTK (Visualisation Toolkit)
- einfaches Interface
- kann auch Interaktiv verwendet werden

- [Swarmintelligence.org](http://www.swarmintelligence.org)
`http://www.swarmintelligence.org`
- [Wikipedia](http://en.wikipedia.org/wiki/Particle_Swarm_Optimization)
`http://en.wikipedia.org/wiki/Particle_Swarm_Optimization`
- [Particle Swarm Central](http://www.particleswarm.info)
`http://www.particleswarm.info`
- [Python](http://www.python.org)
`http://www.python.org`
- [NumPy](http://numpy.scipy.org)
`http://numpy.scipy.org`
- [Mayavi2](http://code.enthought.com/projects/mayavi)
`http://code.enthought.com/projects/mayavi`