

Particle Swarm Optimization

Summary of a paper by
James Kennedy and Russell Eberhart

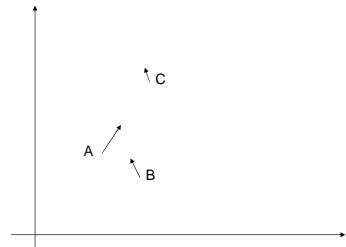
Einleitung

- Particle Swarm Optimization (PSO)
- Eine Methode zum Optimieren von kontinuierlichen nichtlinearen Funktionen
- Hat Verbindungen zu A-life, flocking, und evolutionary computing

Struktur des Systems

- 2D
- Agenten:
 - Durch Zufall bestimmte Plazierung
 - Durch Zufall bestimmte X,Y Geschwindigkeiten
- In jeder Iteration wird fuer jeden Agenten der naechste Nachbar bestimmt und dessen Geschwindigkeit uebernommen.

Struktur des Systems



“Craziness”

- Diese sehr einfache Regel hat dazu gefuehrt das alle Agenten sehr schnell in dieselbe, nichtwechselnde Richtung sich begeben haben.
- Um dem abzuhalten wurden die Geschwindigkeiten durch einen kleinen zufaellig gestimmten Faktor geaendert.
- Dies ergab Variation die aber total kuenstlich war.

Optimierung

- Sinn dieses Systems ist es ein Optimum zu finden.
- Jeder Agent bestimmt den Wert der Funktion an seiner gegenwaertigen Position
- Der beste Wert (pbest[]) und seine X,Y Position werden gespeichert (pbestx[], pbesty[])
- Laenge des Vektoren = Anzahl der Agenten

Benutzung des Lokalen Optimums

- Wenn der Agent rechts seines Optimums ist wird seine Geschwindigkeit verringert:
 $vx[] = vx[] - rand() * p_increment$
- "p_increment" ist ein Parameter
- Wenn der Agent links seines Optimums ist, wird seine Geschwindigkeit erhöht
- Ähnlich für die vertikale Achse

Globales Maximum

- Der bisher beste Wert und seine Position werden separat gespeichert und ist allen Agenten verfügbar.
- Der Index des Agenten mit dem besten Wert wird "gbest" zusätzlich zugewiesen.
- Nun wird die Geschwindigkeit zusätzlich wie folgt geändert (für beide Dimensionen und Richtungen):
$$\begin{aligned} & \text{if } presentx[] > pbestx[gbest] \\ & \text{then } vx[] = vx[] - rand() * g_increment \end{aligned}$$

Ergebnisse

- Durch eine Simulation wurde bestimmt, dass hohe Werte von p_increment und g_increment die Agenten quasi in das Maximum reinsaugt.
- Mit kleinen Werten von p_increment und g_increment "fliegen" die Agenten gemäßlich zum Maximum.

Auswertung

- "pbest": Bisher bester persönlicher Wert
- "gbest": Rekord der Gruppe
- Kein "flocking", da jeder Agent sich nur um sich und das Maximum der Gruppe kümmert.

Verbesserungen

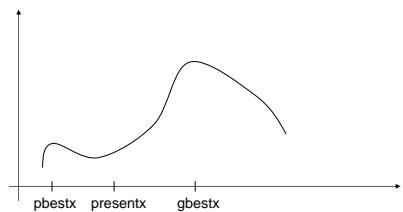
- Die Geschwindigkeit sollte am besten relativ zur Distanz zum Optimum geändert werden:

$$vx[] = vx[] + rand() * p_increment * (pbestx[] - presentx[])$$

Verbesserungen

- In höher-dimensionalen Räumen ist es schwer zu schätzen ob p_increment oder g_increment größer sein soll.
- Anstatt dessen wird der stochastische Faktor mit 2 multipliziert um ihm einen "mean" von 1 zu geben.
- Auf diese Art und Weise werden Agenten das Ziel zur Hälfte "überfliegen."
- Die Formel ist nun:
$$vx[] = vx[] + 2 * rand() * (pbestx[] - presentx[]) + 2 * rand() * (pbest[gbest] - presentx[])$$

Veranschaulichung



Was ist PSO?

- Die Autoren behaupten das “*Conceptually, it seems to lie somewhere between genetic algorithms and evolutionary programming.*”
- Sie behaupten weiterhin das “*Much of the success of PS seems to lie in the agents’ tendencies to hurtle past their target.*”