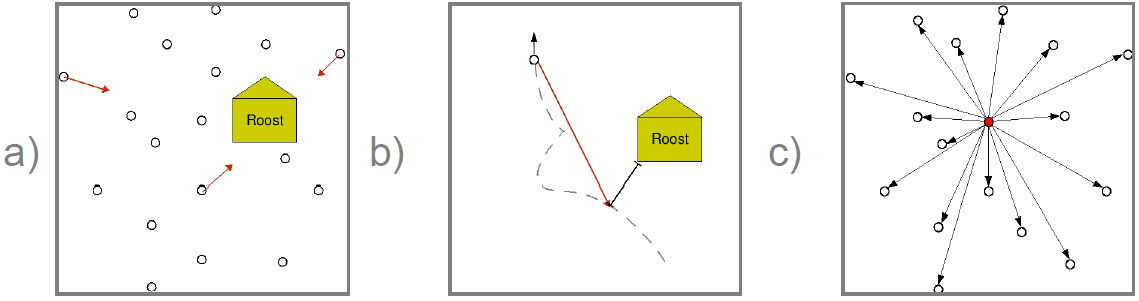
### Schwarmbasierte Optimierung

* Seperation: Agent bewegt sich von Nachbar weg falls zu nah
* Alignment:
* Kohäsion: Agent bewegt sich Richtung Durchschnittsposition seiner Nachbarn
* Die n Individuen kommunizieren einander in/direkt ihre Suchrichtung

**Roost/Schlafplatz**



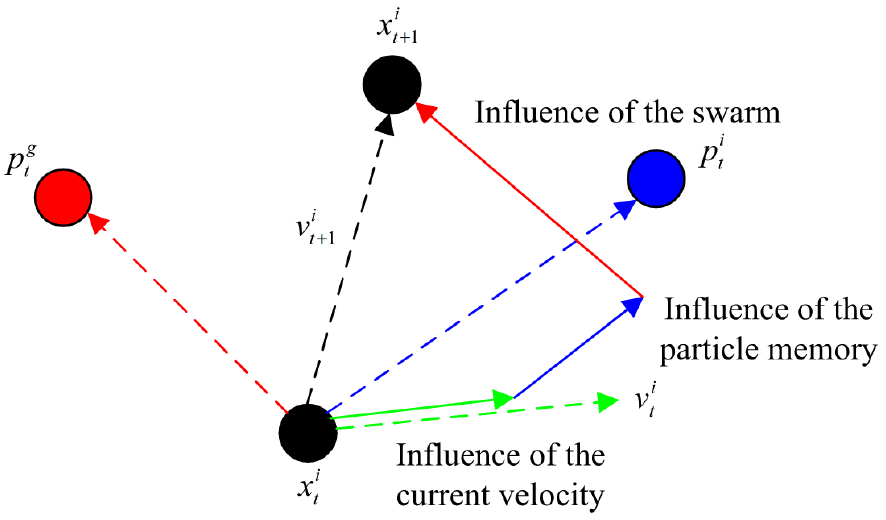
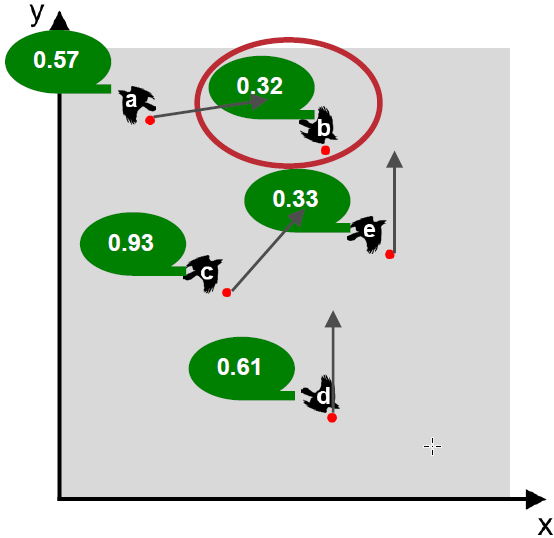
1. Jeden Agent ziehts Richtung Roost
2. Jeder Agent erinnert sich an Location mit geringster Entfernung
3. Jeder Agent teilt diese Location mit Nachbarn

**Anatomie eines Individuums**

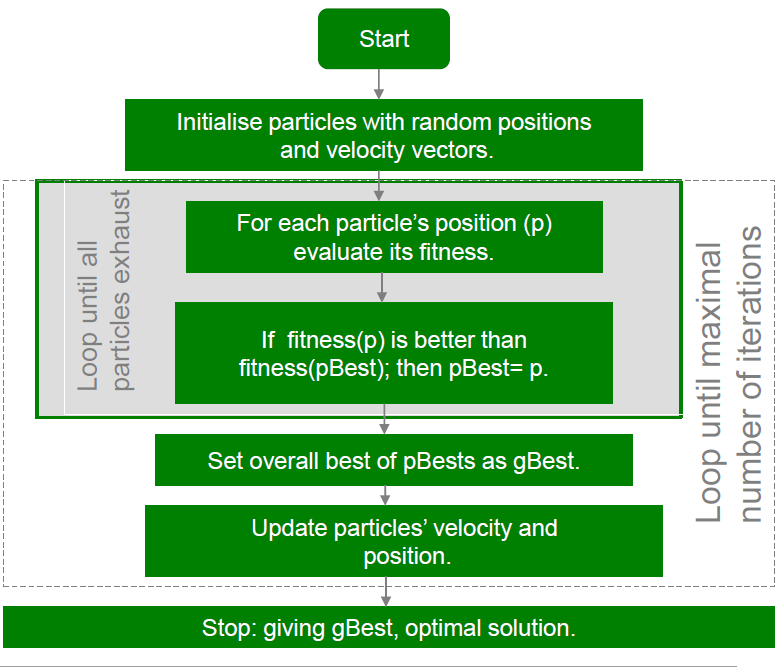
* X-Vektor: Momentane Posi
* P-Vektor: Beste Posi bisher
* V-Vektor: Travelrichtung, falls undisturbed
* X-Fitness: Fitness X Vektor
* P-Fitness: Fitness P Vektor

**Algorithmus**

1. Population (Agenten/Partikel) kreieren und gleichmäßig über x verteilen
2. Positionen evaluieren anhand Zielfunktion
3. Update falls besser als previous best
4. Momentan bestes Partikel bestimmen
5. Update der Partikelgeschwindigkeit



1. Partikel bewegen
2. Step 2 bis Stopkriterium erreicht



**PSO-Parameter**

* Topology: Ring mit 3er Nachbar oder Stern mit global Nachbar
* Velocityvektoren die geupdatet werden
* Impact: Velocity kann sehr groß werden, Performance leidet falls Vmax ungünstig, Inertia Faktor (Trägheit) sorgt für Eingrenzung
* Schwarm- und Nachbargröße: Tradeoff zwischen Lösungsqualität und Kosten: Global Neighbor scheinen in Bezug auf Rechenpower besser
* Updatestrategie: a-/synchron: a macht neu entdeckte Lösungen schneller benutzbar