OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INF FAKULTÄT FÜR INFORMATIK

Intelligente Systeme

Schwarmintelligenz

Prof. Dr. R. Kruse **C. Braune**

{kruse,cbraune}@iws.cs.uni-magdeburg.de

Institut für Wissens- und Sprachverarbeitung
Fakultät für Informatik
Otto-von-Guericke Universität Magdeburg

Schwarm- und populationsbasierte Optimierung

Schwarm-Intelligenz

- Bereich der KI, der intelligente Multi-Agentensysteme entwickelt
- Inspiration durch das Verhalten bestimmter Tierarten, speziell sozialer Insekten (z.B. Ameisen, Termiten, Bienen, etc.) und in Schwärmen lebender Tiere (z.B. Fische, Vögel, etc.)
- Tiere dieser Arten können recht komplexe Aufgaben lösen (Finden von Nahrungsquellen, Wegesuche, Nestbau, etc.), indem sie kooperieren.

Wesentliche Ideen

- Ziemlich einfache Einzelindividuen mit begrenzten Fähigkeiten
- Koordination ohne zentrale Steuerung, lediglich Selbstorganisation
- Austausch von Informationen zwischen Individuen (Kooperation)
- Klassifizierung der Verfahren nach Art des Informationsaustausches

Verfahren

Genetische/Evolutionäre Algorithmen

- Biologisches Vorbild: Evolution der Lebewesen
- Informationsaustausch durch Rekombination der Genotypen
- Jedes Individuum ist ein Lösungskandidat

Verfahren

Teilchenschwarmoptimierung

- Biologisches Vorbild: Futtersuche von Fisch- und Vogelschwärmen
- Informationsaustausch über einfache Aggregation der Einzellösungen
- Jedes Individuum ist ein Lösungskandidat

Ameisenkolonienalgorithmen

- Biologisches Vorbild: Wegesuche zu Futterquellen durch Ameisen
- Informationsaustausch über Veränderung der Umgebung (Stigmergie, erweiterter Phänotyp nach Dawkins)
- Individuen konstruieren Lösungskandidaten

Teilchenschwarmoptimierung

© Eric T. Schulz http://www.eeb.uconn.edu/courses/eeb296/

© Ariel Bravy http://www.skphoton.com/albums/

Fische, Vögel suchen in Schwärmen nach ergiebigen Futterplätzen.
Orientierung anhand individueller Suche (kognitiver Anteil) und an anderen Mitgliedern des Schwarmes in ihrer Nähe (sozialer Anteil)
Außerdem: Leben im Schwarm schützt Individuen gegen Fressfeinde.

Biologische Vorbilder: Fische

©http://s3.amazonaws.com/xlsuite_production/assets/51840/fish-swarm.jpg

2-3 Artgenossen geben einzelnem Fisch im Schwarm Orientierung

Anpassung von Geschwindigkeit und Richtung, sodass Winkel zwischen Schwimmrichtung und nächsten Artgenossen: 45–135 °

Andere Fische werden ignoriert

Garantie, dass Schwarm ständig vorwärtsschwimmt und Abweichler nicht Gesamtrichtung beeinflussen

Teilchenschwarmoptimierung

Motivation: Verhalten von z.B. Fischschwärmen bei der Futtersuche: zufälliges Ausschwärmen, aber stets auch Rückkehr zum Schwarm, Informationsaustausch zwischen Mitgliedern

Ansatz: verwende statt nur einzelnen aktuellen Lösungskandidaten „Schwarm“ von m Lösungskandidaten

Voraussetzung: $\Omega \subseteq \mathbb{R}^n$ und somit zu optimierende (o.B.d.A.: zu maximierende) Funktion $f : \mathbb{R}^n \to \mathbb{R}$

Vorgehen: fasse jeden Lösungskandidaten als „Teilchen“ auf, das Ort $\boldsymbol{x}_i$ im Suchraum und Geschwindigkeit $\boldsymbol{v}_i$ hat ($i = 1, \ldots, m$)

Vereinigt Elemente der bahnorientierten Suche (z.B. Gradientenverfahren) und populationsbasierter Suche (z.B. EA)

Teilchenschwarmoptimierung

Aktualisierung für Ort und Geschwindigkeit des i-ten Teilchens:

$$\boldsymbol{v}_i(t+1) = \alpha \boldsymbol{v}_i(t) + \beta_1 \left(\boldsymbol{x}_i^{(\text{lokal})}(t) - \boldsymbol{x}_i(t)\right) + \beta_2 \left(\boldsymbol{x}^{(\text{global})}(t) - \boldsymbol{x}_i(t)\right)$$

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t)$$

Parameter: β_1, β_2 zufällig in jedem Schritt, α mit t abnehmend

$\boldsymbol{x}_i^{(\text{lokal})}$ ist **lokales Gedächtnis** des Individuums (Teilchens): der beste Ort im Suchraum, den Teilchen bisher besucht hat, d.h.

$$\boldsymbol{x}_i^{(\text{lokal})} = \boldsymbol{x}_i \left(\arg\max_{u=1}^{t} f(\boldsymbol{x}_i(u))\right)$$

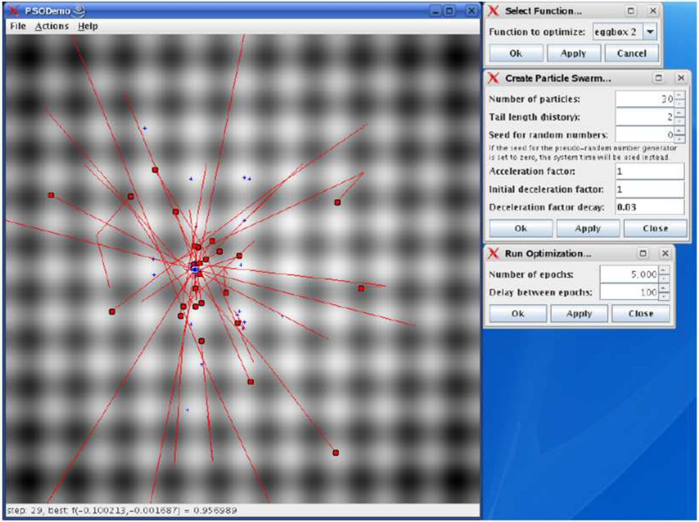
$\boldsymbol{x}^{(\text{global})}$ ist **globales Gedächtnis** des Schwarms: der beste Ort im Suchraum, den Individuum des Schwarms bisher besucht hat (beste bisher gefundene Lösung), d.h.

$$\boldsymbol{x}^{(\text{global})}(t) = \boldsymbol{x}_j^{(\text{lokal})}(t) \quad \text{mit} \quad j = \arg\max_{i=1}^{m} f\left(\boldsymbol{x}_i^{(\text{lokal})}\right)$$

Algorithmus 1 Teilchenschwarmoptimierung

for each Teilchen i $\{$
  $\boldsymbol{x}_i \leftarrow$ wähle zufällig im Suchraum Ω
  $\boldsymbol{v}_i \leftarrow 0$
$\}$
do $\{$
  for each Teilchen i $\{$
    $y \leftarrow f(\boldsymbol{x}_i)$
    if $y \geq f\left(\boldsymbol{x}_i^{(\text{lokal})}\right)$ $\{$
      $\boldsymbol{x}_i^{(\text{lokal})} \leftarrow \boldsymbol{x}_i$
    $\}$
    if $y \geq f\left(\boldsymbol{x}_i^{(\text{global})}\right)$ $\{$
      $\boldsymbol{x}^{(\text{global})} \leftarrow \boldsymbol{x}_i$
    $\}$
  $\}$
  for each Teilchen i $\{$
    $\boldsymbol{v}_i(t+1) \leftarrow \alpha \cdot \boldsymbol{v}_i(t) + \beta_1 \left(\boldsymbol{x}_i^{(\text{lokal})}(t) - \boldsymbol{x}_i(t)\right) + \beta_2 \left(\boldsymbol{x}^{(\text{global})}(t) - \boldsymbol{x}_i(t)\right)$
    $\boldsymbol{x}_i(t+1) \leftarrow \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t)$
  $\}$
$\}$ **while** Terminierungskriterium ist nicht erfüllt

Teilchenschwarmoptimierung



Ameisenkolonieoptimierung

© PeTA http://www.helpingwildlife.com/ants.asp

© NickLyonMedia http://nicklyon.orchardhostings4.co.uk

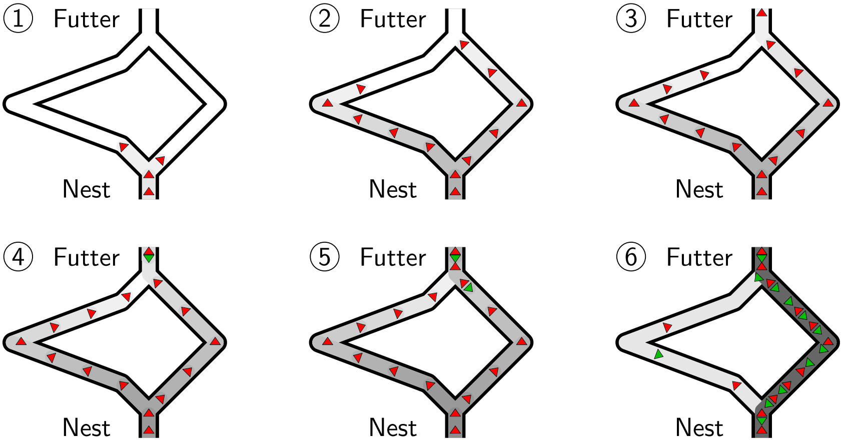
Da gefundenes Futter zur Versorgung der Nachkommen zum Nest transportiert werden muss, bilden Ameisen Transportstraßen

Weglängen zu Futterplätzen werden annähernd minimiert

Doppelbrückenexperiment

- Ameisennest und Futterquelle werden durch Doppelbrücke verbunden (beiden Zweige der Brücke sind verschieden lang)
- Experiment mit argentinischer Ameise *Iridomyrmex Humilis*: diese Ameisenart ist (wie fast alle anderen auch) fast blind (Ameisen können also nicht sehen, welcher Weg kürzer ist)
- Schon nach wenigen Minuten: in meisten Versuchen benutzten fast alle Ameisen kürzeren Weg

Doppelbrückenexperiment



Doppelbrückenexperiment

- Auf kürzerem Weg erreichen Ameisen Futter schneller (Ende des kürzeren Weges erhält daher (am Anfang) mehr Pheromon)
- Auf Rückweg wird wegen entstandener Pheromondifferenz mit höherer W'keit kürzerer Weg gewählt (führt zu einer Verstärkung der Pheromondifferenz)

Ameisenkolonieoptimierung

Motivation: Ameisen einiger Arten finden kürzeste Wege zu Futterquellen durch Legen und Verfolgen von Pheromon („Duftmarken“)

- Intuitiv: kürzere Wege erhalten in gleicher Zeit mehr Pheromon
- Wege werden zufällig nach vorhandenen Pheromonmenge gewählt (es ist um so wahrscheinlicher, dass ein Weg gewählt wird, je mehr Pheromon sich auf Weg befindet)
- Menge des ausgebrachten Pheromons kann von Qualität und Menge des gefundenen Futters abhängen

Grundprinzip: Stigmergie (engl. stigmergy)

- Ameisen kommunizieren indirekt über Pheromonablagerungen
- Stigmergie (indirekte Kommunikation durch Veränderung der Umgebung) ermöglicht global angepasstes Verhalten aufgrund lokaler Informationen

Natürliche und künstliche Ameisen

abstrahiere zu Suche nach bestem Weg in gewichtetem Graphen

- **Problem:** Kreise, die sich selbst verstärken (durchläuft Ameise Kreis, erzeugt sie durch abgelegtes Pheromon Tendenz, Kreis erneut zu durchlaufen)
- **Abhilfe:** Ablegen von Pheromon erst nach Konstruktion des ganzen Weges (Entfernen von Kreisen, bevor Pheromon abgelegt wird)
- **Problem:** Ggf. konzentriert sich Suche auf am Anfang konstruierte Lösungskandidaten (vorzeitige Konvergenz)
- **Abhilfe:** Pheromonverdunstung (spielt in Natur geringe Rolle)

Nützliche Erweiterungen/Verbesserungen

- Abgelegte Pheromonmenge hängt von Lösungsgüte ab
- Einbringen von Heuristiken in Kantenwahl (z.B. Kantengewicht)

Ameisenkolonieoptimierung

Voraussetzungen: kombinatorisches Optimierungsproblem mit konstruktiver Methode, zur Erzeugung eines Lösungskandidaten

Vorgehen: Lösungen werden durch Folge von Zufallsentscheidungen konstruiert, wobei jede Entscheidung Teillösung erweitert

Entscheidungsfolge = Pfad in Entscheidungsgraphen (auch: Konstruktionsgraphen)

Ameisen sollen Pfade durch Entscheidungsgraphen erkunden und besten (kürzesten, billigsten) Weg finden

Ameisen markieren benutzte Kanten des Graphen mit Pheromon
$\Rightarrow$ andere Ameisen werden zu guten Lösungen geleitet

Pheromon verdunstet in jeder Iteration, damit einmal ausgebrachtes Pheromon System nicht zu lange beeinflußt („Vergessen" veralteter Information)

Anwendung auf TSP

Darstellung des Problems durch $n \times n$ Matrix $\boldsymbol{D} = (d_{ij})_{1 \le i,j \le n}$

n Städte mit Abständen d_{ij} zwischen Städte i und j

Beachte: $\boldsymbol{D}$ kann asymmetrisch sein, aber $\forall i \in \{1, \ldots, n\} : d_{ii} = 0$

Pheromoninformation als $n \times n$ Matrix $\Phi = (\phi_{ij})_{1 \le i,j \le n}$

Pheromonwert $\phi_{ij} (i \neq j)$ gibt an, wie wünschenswert es ist, Stadt j direkt nach Stadt i zu besuchen (ϕ_{ii} nicht benötigt)

Φ muss nicht notwendig symmetrisch sein/gehalten werden

Alle ϕ_{ij} werden mit gleichen kleinen Wert initialisiert (anfangs liegt auf allen Kanten gleiche Menge Pheromon)

Ameisen durchlaufen (durch Pheromon) Hamiltonkreise (sie markieren Kanten des durchlaufenden Hamiltonkreises mit Pheromon, wobei ausgebrachte Pheromonmenge der Lösungsqualität entspricht)

Lösungskonstruktion

Jede Ameise hat „Gedächtnis“ C, welche Indizes der noch nicht besuchten Städte enthält

Jede besuchte Stadt wird aus Menge C entfernt

Gedächtnis gibt es im biologischen Vorbild nicht!

Ameise wird in zufällig bestimmter Stadt gesetzt (Anfang der Rundreise)

Ameise wählt noch nicht besuchte Stadt und begibt sich in diese: in Stadt i wählt Ameise (unbesuchte) Stadt j mit W'keit

$$p_{ij} = \frac{\phi_{ij}}{\sum_{k \in C} \phi_{ik}}.$$

Wiederhole Schritt 2 bis alle Städte besucht

Pheromonaktualisierung

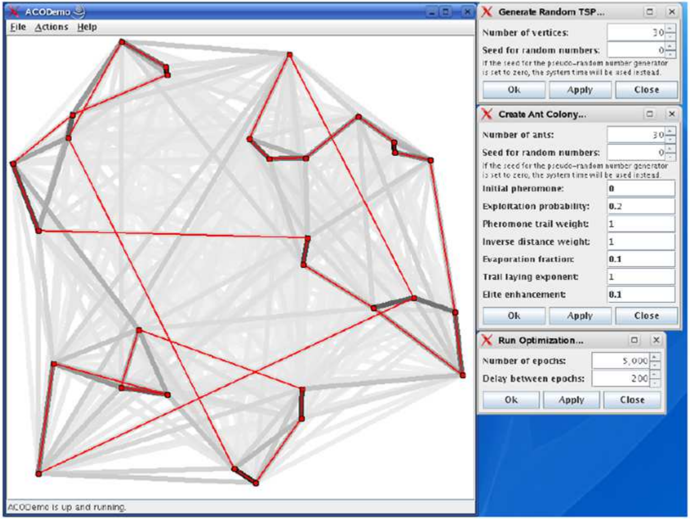
1. **Verdunstung/Evaporation**
   alle ϕ_{ij} werden um Bruchteil η (evaporation) verringert:
   $$\forall i, j \in \{1, \ldots, n\} : \phi_{ij} = (1 - \eta) \cdot \phi_{ij}$$
2. **Verstärkung konstruierter Lösungen**
   Kanten der konstruierten Lösungen werden mit zusätzlicher Menge an Pheromon belegt, die Lösungsqualität entspricht:
   $$\forall \pi \in \Pi_t : \phi_{\pi(i)\pi((i \bmod n)+1)} = \phi_{\pi(i)\pi((i \bmod n)+1)} + Q(\pi)$$
   Π_t ist Menge der im Schritt t konstruierten Rundreisen (Permutationen), Qualitätsfunktion: z.B. inverse Reiselänge
   $$Q(\pi) = c \cdot \left(\sum_{i=i}^{n} d_{\pi(i)\pi((i \bmod n)+1)} \right)^{-1}$$
   „Je besser Lösung, desto mehr Pheromon erhalten deren Kanten."

Problem des Handlungsreisenden

Algorithmus 2 Ameisenkolonieoptimierung für das TSP

initialisiere alle Matrixelemente ϕ_{ij}, $1 \leq i, j \leq n$, auf kleinen Wert ϵ
do $\{$
    for each Ameise $\{$ /* konstruiere Lösungskandidaten */
        $C \leftarrow \{1, \ldots, n\}$ /* Menge der zu besuchenden Städte */
        $i \leftarrow$ wähle zufällig Anfangsstadt aus C
        $C \leftarrow C \setminus \{i\}$ /* entferne sie aus den unbesuchten Städten */
        while $C \neq \emptyset$ $\{$ /* solange nicht alle Städte besucht wurden */
            $j \leftarrow$ wähle nächste Stadt der Reise aus C mit W'keit p_{ij}
            $C \leftarrow C \setminus \{j\}$ /* entferne sie aus den unbesuchten Städten */
            $i \leftarrow j$ /* und gehe in die ausgewählte Stadt */
        $\}$
    $\}$
    aktualisiere Pheromon-Matrix Φ nach Lösungsgüte
$\}$ **while** Terminierungskriterium ist nicht erfüllt

Ameisenkolonieoptimierung



Allg. Anwendung zur Optimierung

Grundsätzliches Prinzip
Formuliere Problem als Suche in (Entscheidungs-)Graphen, Lösungskandidaten müssen durch Kantenmengen beschreibbar sein, (beachte: es muss sich nicht notwendigerweise um Pfade handeln!)

Allgemeine Beschreibung: im folgenden jeweils angegeben:

- Knoten, Kanten des Entscheidungs-/Konstruktionsgraphen
- Einzuhaltende Nebenbedingungen
- Bedeutung des Pheromons auf Kanten (und evtl. Knoten)
- Nutzbare heuristische Hilfsinformation
- Konstruktion eines Lösungskandidaten

Algorithmisches Vorgehen ist i.W. analog zu Vorgehen beim TSP

Allg. Anwendung zur Optimierung: TSP

Knoten und Kanten des Entscheidungs-/Konstruktionsgraphen: die zu besuchenden Städte und ihre Verbindungen, die Verbindungen sind gewichtet (Abstand, Zeit, Kosten)

einzuhaltende Nebenbedingungen: besuche jede Stadt genau 1x

Bedeutung des Pheromons auf den Kanten: wie wünschenwert ist es, Stadt j nach Stadt i zu besuchen

nutzbare heuristische Hilfsinformation: Abstand der Städte, bevorzuge nahe Städte

Konstruktion eines Lösungskandidaten: ausgehend von zufällig gewählter Stadt wird stets zu einer weiteren, noch nicht besuchten Stadt fortgeschritten

Allg. Anwendung zur Optimierung: Zuordnungsproblem

Ordne n Aufgaben zu m Arbeiter (Personen, Maschinen): Minimierung der Summe der Zuordnungskosten d_{ij} unter Einhaltung maximaler Kapazitäten ρ_j bei gegebenen Kapazitätskosten r_{ij}, $1 \leq i \leq n$, $1 \leq j \leq m$

- Jede Aufgabe und jeder Arbeiter = Knoten des Konstruktionsgraphen (Kanten tragen die Zuordnungskosten d_{ij})
- Jede Aufgaben muss genau einem Arbeiter zugeordnet werden (Kapazitäten der Arbeiter nicht überschreiten)
- Pheromonwerte auf Kanten beschreiben, wie wünschenswert Zuordnung einer Aufgabe an Arbeiter ist
- Inverse absolute oder relative r_{ij} oder inverse d_{ij}
- Wähle schrittweise Kanten (müssen keinen Pfad bilden), übergehe Kanten von bereits zugeordneten Aufgaben (bestrafe Lösungen, die Nebenbedingungen verletzen (Kostenerhöhung))

Allg. Anwendung zur Optimierung: Rucksackproblem

Wähle aus n Objekten mit zugeordnetem Wert w_i, Gewicht g_i, Volumen v_i, etc. $1 \leq i \leq n$, Teilmenge maximalen Wertes aus, sodass Maximalwerte für Gewicht, Volumen, etc. eingehalten

- Jedes Objekt = Knoten des Konstruktionsgraphen (Knoten tragen Objektwerte w_i, Kanten nicht benötigt)
- Maximalwerte für Gewicht, Volumen, etc. müssen eingehalten werden
- Pheromonwerte: nur Knoten zugeordnet (sie beschreiben, wie wünschenswert Auswahl des zugehörigen Objektes)
- Verhältnis von Objektwert zu relativem Gewicht, Volumen, etc. wobei in den Verhältnissen die Maximalwerte berücksichtigt werden können
- Wähle schrittweise Knoten aus, wobei in jedem Schritt sichergestellt, dass Maximalwerte eingehalten

Konvergenz der Suche

betrachte „Standardverfahren“ mit folgenden Eigenschaften:

- Verdunstung des Pheromons mit konstantem Faktor von allen Kanten
- Nur auf Kanten des besten, bisher gefundenen Lösungskandidaten wird Pheromon abgelegt (strenges Eliteprinzip)
- $\exists$ Untergrenze ϕ_{min} für Pheromonwerte der Kanten, welche nicht unterschritten wird
- Standardverfahren konvergiert in W'keit gegen Lösung, d.h. mit $t \to \infty$ geht W'keit, dass Lösung gefunden wird, gegen 1
- Lässt man Untergrenze ϕ_{min} für Pheromonwerte „genügend langsam“ gegen 0 gehen ($\phi_{\text{min}} = \frac{c}{\ln(t+1)}$ mit Schrittzahl t und Konstante c), kann man zeigen, dass für $t \to \infty$ jede Ameise der Kolonie Lösung mit gegen 1 gehender W'keit konstruiert

Zusammenfassung

- Schwarm- und populationsbasierte Algorithmen: **Heuristiken zur Lösung von Optimierungsproblemen**
- Ziel: Finden guter Näherungslösungen
- **Problem der lokalen Optima:** Reduzierung durch bessere Durchforstung des Suchraums
- Wichtig: **Informationsaustausch** zwischen Individuen (je nach Prinzip: verschiedene Algorithmentypen)
- **Teilchenschwarmoptimierung**
  - Optimierung einer Funktion mit reellen Argumenten
  - Informationsaustausch durch Orientierung an Nachbarn
- **Ameisenkolonieoptimierung**
  - Suche nach besten Wegen (abstrakt: in einem Entscheidungsgraphen)
  - Informationsaustausch durch Veränderung der Umgebung (Stigmergie)

Literatur

Birattari, M., Di Caro, G. A., Doursat, R., Engelbrecht, A. P., Floreano, D., Gambardella, L. M., Groß, R., Sahin, E., Stützle, T., and Sayama, H. (2010).
Swarm Intelligence.
Springer.

Clerc, M. (2010).
Particle Swarm Optimization.
John Wiley & Sons.

Goss, S., Aron, S., Deneubourg, J.-L., and Pasteels, J. M. (1989).
Self-organized shortcuts in the argentine ant.
Naturwissenschaften, pages 579–581.

Kruse, R., Borgelt, C., Braune, C., Moewes, C., and Steinbrecher, M. (2015).
Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze.
Springer Vieweg, Wiesbaden.