

Notizen zum Skype Meeting vom 19.11.2015

von Basti und Sarah

26. November 2015

Kapitel 1

Distributed Recognition

1.1 Allgemein/Ablauf

- Flos Arbeit über nestmate recognition als Vorarbeit
- gemeinsame VÖ über TruDiPhe Anfang 2016 (Frage: Ideen für Konferenz?)
- dann DFG-Antrag für 2 Arbeiter

1.2 Inhalte

2 Hauptansatzpunkte möglich:

1. Nestmate recognition (NR)

- erkennen als freundlich → positiv
- bekämpfen/feindlich → negativ

Aber: nicht feiner differenzierbar, da Ameisen sich in der Situation, wenn sie anderer Ameise gegenüberstehen, sofort entscheiden müssen, ob sie angreifen oder nicht.

2. Footprints (FP)

für Flo interessanter, da facettenreicher (Spatial Patterns through footprint response?). Ameisen können Fußspuren

- folgen → positiv
- ignorieren → neutral
- meiden → negativ

Mehr Möglichkeiten, Verhalten abzuwägen zB auch starker/schwacher Drang, den Fußspuren zu folgen.

Ausgangslage Basti/Sarah

3 Teile:

- **Trusted Desktop Grid:** Simulation von Multi-Agenten-System, abstrakte Jobs, die in Work Units gesplittet werden und verteilt gerechnet werden. Ziel ist, speedup zu maximieren (von einzelner Agenten und Gesamtsystem) $speedup = \frac{time_{self}}{time_{distributed}}$, also: wie lang brauch ich wenn ich alles selber rechne im Vergleich zu wenn ichs verteile?. Natürlich müssen die Agenten dann aber auch für andere arbeiten, um sich Vertrauen zu verdienen, denn sonst nimmt Dir auch keiner Deine Arbeit ab. Es gibt Trust Communities (TCs), Zusammenschluss von sich gegenseitig stark vertrauenden Agenten
- **Trusted Ants:** Simulation in Unity, TDG in "schön"/ nachvollziehbar/ leichter zu verstehen; hier sind Jobs Wegpunkte, die abgelaufen werden müssen
- **T-Grid:** Real-Implementierung des TDG, hoffentlich im Februar in erster, lauffähiger Prototyp. Ist Software, die auf mehrere Rechner gespielt wird, diese können sich dann, wenn sie vertrauen, gegenseitig Rechenleistung abtreten, wenn sie grad nicht genutzt wird.

Problem bei allen drei: zentrale Datenbank für Reputationswerte (Reputation $\hat{=}$ aggregierter Trust). Z.B. in TC bei Rust Community Manager (TCM) Als erster Lösungsansatz gibt es das Konzept der **Digitalen Trust Pheromone** (TruDiPhe), d.h., Trust-Werte werden bei dem jeweiligen Adressat dieses Wertes verschlüsselt gespeichert.

Fragen/ Zum Nachdenken

Eher Basti/Sarah:

Wie mappen wir das alles auf NR/FP?

Mappt Job**annahme** auf FP? Sollen wir die Effektivität klassifizieren? Also bei T-Grid, so dass das Grid gut ausgelastet und zielführend (wichtige Sachen zuerst?) benutzt wird? Erweiterung der Überlegungen auf Arbeitgeben (submitter)! Prios einführen bei Arbeiten!

Also: Mappt „Kriterium zielführend“ auf „Art der FP“? Mapping Pfad \leftrightarrow Aufgabe

NR: wie Schablone, die man anpasst \rightarrow zu trivial

Radius eingrenzen \leftrightarrow Größe der TC

Lernen! anfangs misstrauisch am besten? Vertrauensvorschuss gewähren?

Eher Flo:

Zusammenhang FP und Umwelteinflüsse?

Neugier modellieren?

Wie schnell verschwindet hydrocarbon? Ameisen nehmen Stärke FP in Entscheidung auf, aber können nicht unterscheiden ob neu und schwach oder alt und intensiv gewesen.

Überlagerung von FP

FP ignorieren/alleine Futter finden \leftrightarrow submitter können auch selber rechnen (B&S)

bei Vertrauensbruch Fußspuren weniger folgen / weniger explorativ

Beide:

Arbeitsvergabe (B&S) \leftrightarrow Suche der Fußspuren (F) !!! (X)

Systemperformanz als Metrik?

Evolution - Nachkommen erben Parameter

In offenem System selbst-organisierte Effektivität; ohne zentrale Kontrolle beste Pfade zu beschreiben rausfinden

Optimale Strategie finden, die wir dann anwenden. In TDG: pro Agentenzusammensetzung optimale Strategie zur Laufzeit lernen. quantitativ: Parametergrenzen identifizieren und allgemein was daraus ableiten;

1.3 ToDo

- working title finden
- Gemeinsame Zielsetzung formulieren
- Zeitplan erstellen
- ...?