Koordination in MAS

Seminar: Verteilte Prozesse

Wilfried Teiken
wteiken@cl-ki.uni-osnabrueck.de

4. Nov 1999

Koordination

Prozessschritt	**Beispiel**
Problemanalyse	Refinement in Agent Groups
Kompetenzabschätzung	To Learn or Not to Learn
Teambildung	
Aufgabenverteilung	Using Reciprocity to Adapt to Others
Ressourcenzuteilung	
Konfliktresolution	

Verteilte Problemanaylse

Bei der verteilten Problemanalyse gibt es einige Probleme:

- Wissen liegt verteilt bei verschiedenen Agenten
- Wissen ist unterspezifiziert
- Repräsentationsformalismen sind unterschiedlich

Trotz dieser Probleme soll sich das Gesamtsystem kohärent verhalten und effizient eine Problemlösung finden.

Verteilte Problemanalyse - Vorraussetzungen

Jeder Agent besitzt wissen über:

Kommunikation: Eine Sprache, Kommunikationsmechanismen (KIF, KQML)

Interaktion: Methoden zur Koordination (Contract Net)

Agentenmodelle: Modellierungen der internen Zustände anderer Agenten

Verteilte Problemanalyse - Verteilte Planung

Jeder der Agenten kann durch einen Fehler den Gesamtplan gefährden. Fehler entstehen durch:

Inkorrekte Daten: Das (Welt-)Wissen ist inkorrekt (im gegebenen Kontext)

Unvollständige Daten: Es fehlt Wissen über eine Problemlösung

Inkonsistente Daten: Verschiedene Agenten haben inkonsistente Annahmen

Unzureichende Daten: Aufgrund der gegebenen Daten(verteilungen) ist die Problemlösung zu Aufwendig.

Verteilte Problemanalyse - Das Beispielszenario

- Einfaches *Hunter-Prey*-Szenario
- *Hunter* können zur Jagd grosser *Preys* kooperieren
- einfacher Zyklus: Opfer suchen, zum Opfer laufen, Opfer töten, Opfer fressen
- Sichtverhältnisse und andere Umgebungsparameter werden vom Umweltagenten festgelegt
- Das Szenario soll für spätere Tests verfeinert werden

Verteilte Problemanalyse - Die Agenten

- Jeder Agent besitzt *capabilities*, *beliefs* und *commitments*
- Ein Agent kann ein *group goal* ausschreiben, zu dem sich andere Agenten verpflichten
- Bei einem *commitment* zu einem Gruppenziel muß jeder ausser dem initiierenden Agenten sein persönliches Ziel aufgeben
- Bei Gruppenzielerfüllung (oder der Feststellung der Nichterfüllbarkeit) wird die Gruppe aufgeloßt und die persönlichen Ziele werden reaktiviert

⇒ Agenten müssen sich vom Gruppenziel einen Vorteil versprechen.

Verteilte Problemanalyse - Aktionen

Jede Aktion besitzt *constraint sets* und *immediate preconditions*:

constraints(eat,

time($-$),

[prey($-$)], $-$,

other-request($-$),

p$recond$([$Prey$],

[[believes($hunter1$, [Now, object-position($hunter1$, $Here$)], t),

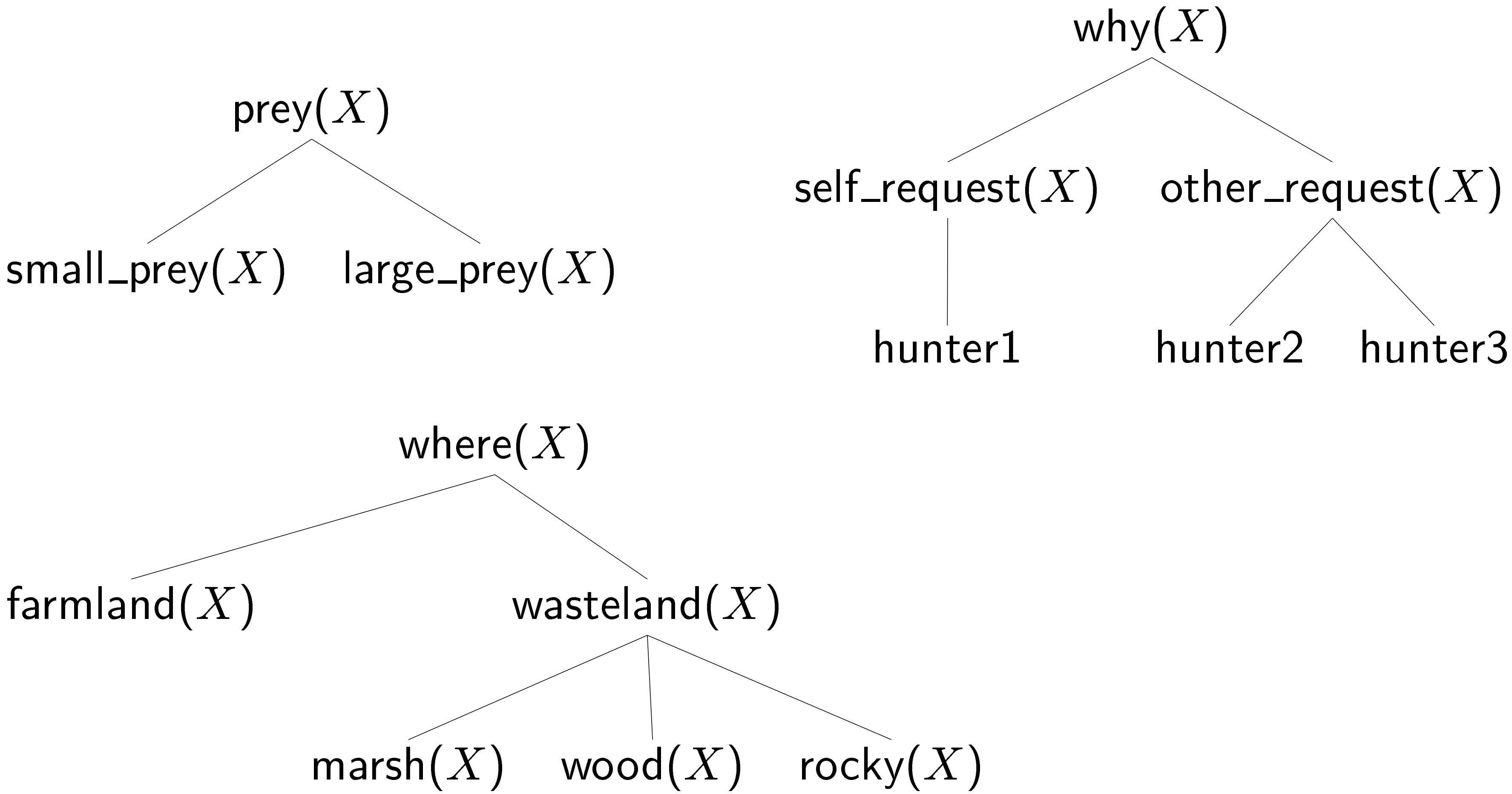
believes($hunter1$, [Now, object-position($Prey$, $Here$)], t),

believes($hunter1$, [Now, object-status($Prey$, $dead$)], t)]]))

Verteilte Problemanalyse - *Refinement*-Operatoren

1. Ein *constraint set* spezialisieren
2. Ein *constraint set* löschen
3. Ein *constraint set* generalisieren
4. Ein *constraint set* hinzufügren
5. Eine *precondition* löschen
6. Eine *precondition* hinzufügen

Verteilte Problemanalyse - Domänenwissen



Verteilte Problemanalyse - Refinement

- where(X)
  - farmland(X)
  - wasteland(X)
    - marsh(X)
    - wood(X)
    - rocky(X)

constraint(next-loc, time(-),
[wasteland(-)], -, why(-),
precond([Loc], []))

$\Downarrow$ $\Downarrow$

- where(X)
  - farmland(X)
  - wasteland(X)
    - marsh(X)
    - new1(X)
      - wood(X)
      - rocky(X)

constraint(next-loc, time(-),
[new1(-)], -, why(-),
precond([Loc], []))

Kompetenzabschätzung

- Kein Agent besitzt vollständiges Wissen über die anderen Agenten
- Ein Agent kann eine Aufgabe nicht komplett Überblicken
- Der Agent muß sich Agenten suchen, die für Ihn (Teil-)Probleme lösen
- Der Agent muß anhand unvollständigen Wissens eine Aufgabe vergeben
- Der Agent sollte lernen, welche Agenten welche Aufgaben lösen können (konnten)

Kompetenzabschätzung - Das Beispielszenario

- Ein *Problem Solving Environment* für mathematische Aufgaben (PDG)
- Es sollen PDG's, die aus Anwendungsspezischen Modellen entstehen, gelöst werden
- Der Systembenutzer soll die Aufgaben ohne tieferes Verständnis für die numerischen Methoden lösen können
- Es werden Vorschläge für die Loßungsmethode (zusammen mit einer Wahrscheinlichkeitsangabe) zu einer PDG abgegeben
- Die Vorschläge werden durch CBR aus einer Datenbank entnommen

Kompetenzabschätzung - Das Agentensystem

- Jeder Agent gehört zu einem Wissenschaftler
- Da jeder Wissenschaftler ein spezifisches Arbeitsgebiet hat, enthalten die Datenbasen unterschiedlich gelagerte Problemecluster
- Wenn ein Agent keine (ausreichend wahrscheinliche) Loßung findet, dann versucht er andere Agenten zu befragen
- Jeder Agent lernt eine Zuordnung von den Problemgruppen zum spezialisierten Agenten

Kompetenzabschätzung - Der Lernprozess

- Das System braucht klassifizierte Daten, um die Zuornung lernen zu können

- Der Benutzer kann dem System keine Daten zur Klassifikation liefern, da er die Klassifikation gerade *nicht* leisten kann bzw. soll

- Jeder Agent liefert Zusammen mit seiner Lösung für q eine Wahrscheinlichkeit $p(q)$ und einen Nutzenfunktion $U_t(q)$ bzw. $U_f(q)$ Die Nutzenfunktionen U_t und U_f sind definiert als:

$$U_t(q) = -U_f(q) = f(N_e), f(x) = \frac{2}{2 + e^{-x}} - 1, N_e = |\text{Cluster}_q|$$

Kompetenzabschätzung - Der Lernprozess II

- Der mögliche Nutzen ist so definiert als:

$$r(q) = p(q)U_t(q) + p(\neg q)U_f(q), p(\neg q) = 1 - p(q)$$

- Ein Agent nimmt eine Zuordnug auf, wenn der mögliche Nutzen $r(q) > 0$

- Ein Agent lernt, bis er eine ausreichende Zuordnung von Problemclustern zu Agenten gefunden hat

- Spätere Kompetenzneueinschätzungen sollen getriggert werden können

Aufgabenverteilung

- Teams können Aufgaben effizienter bewältigen als Einzelagenten
  - Koordination der Teammitglieder (Kooperationsbereitschaft)
  - optimale Aufteilung der Gesamtaufgabe unter den Mitgliedern
- Teams (aus unterschiedlichen Agenten) ermöglichen erst die Bewältigung komplexer Aufgaben
  - verschiedene Fähigkeiten können kombiniert werden
  - das Gesamtsystem hat eine grössere Kompetenz

Aufgabenverteilung - Das Beispielszenario

- Ein einfacher Lieferdienst
- Zentrales Paketdepot, Ziele radial von der Zentrale ausgehend
- Boten können sich nur entlang der radialen Achsen bewegen
- Boten können untereinander Paketaufträge übernehmen
- Es gibt unterschiedlich kooperative bzw. egoistische Agententypen

Fragestellung: Ist Kooperation in diesem Szenario sinnvoll ?

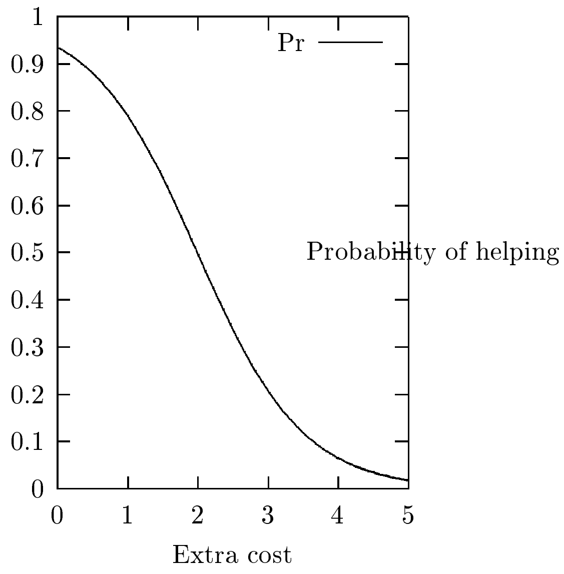
Aufgabenverteilung - Kooperationsbereitschaft

- Eine einzelentscheidung für Kooperation ist i.d.R. unsinnig

- Die Kooperationsbereitschaft wird errechnet als:

$$Pr(i,k,j,l) = \frac{1}{1 + \exp(\frac{C_{ij}^{kl} - \beta C_{\mathsf{avg}}^{k} - B_{ki}}{\tau})}$$

- β und τ bestimmen *Kooperationsbereitschaft*, B_{ki} ist das *Kooperationsgedächnis*



Aufgabenverteilung - Das Beispielszenario II

- Die Kosten für einen Transport sind:
  - *Standardpreis* ist die doppelte Entfernung Zentrale - Ziel
  - Jedes weitere Paket kostet eine Einheit
  - Jeder Extraweg für einen Agenten kostet $2\times$ Distanz eigenes/fremdes Ziel
- Ein Agent übernimmt nur dann Transporte, wenn er selbst etwas in die Richtng bringt
- Ein Agent übernimmt für jeden anderen Agent nur jeweils max. 1 Paket
- Ein Agent übernimmt nur dann ein Paket, wenn der fremde Agent mehr spart als ihn die Aktion kostet

Aufgabenverteilung - Die Agententypen

Es gibt in dem Szenario mehrere Agenten mit unterschiedlicher Kooperationsbereitschaft:

Philanthropische Agenten: übernehmen **immer** Pakete für andere Agenten

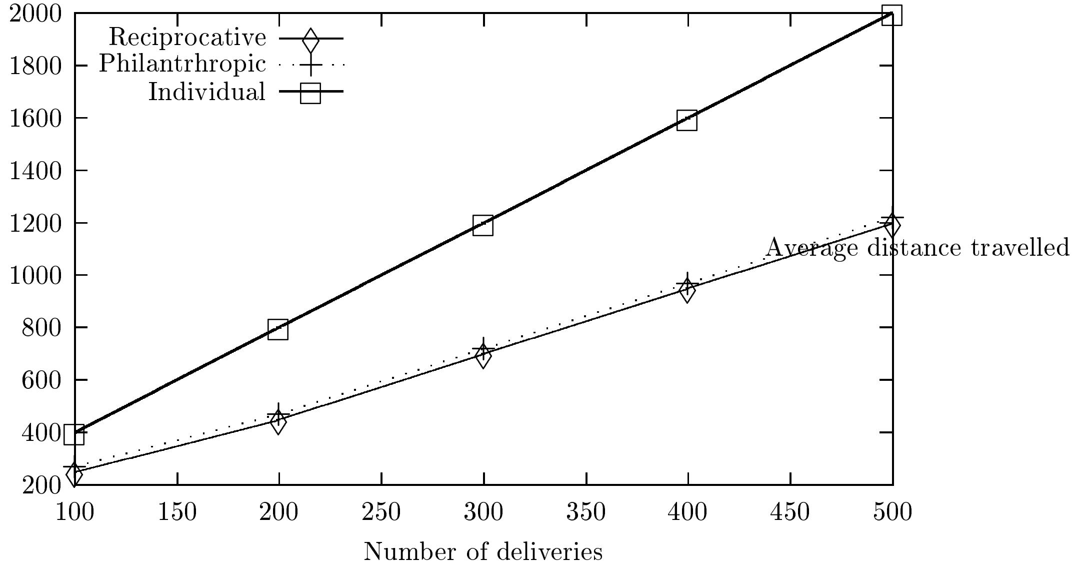
Egoistische Agenten: lassen sich helfen, helfen aber niemals anderen

Kooperierende Agenten: Sie besitzen die vorgestellten stochastischen Kooperationsbereitschaft Pr

Individuelle Agenten: führen ihre Aufträge immer selbst aus

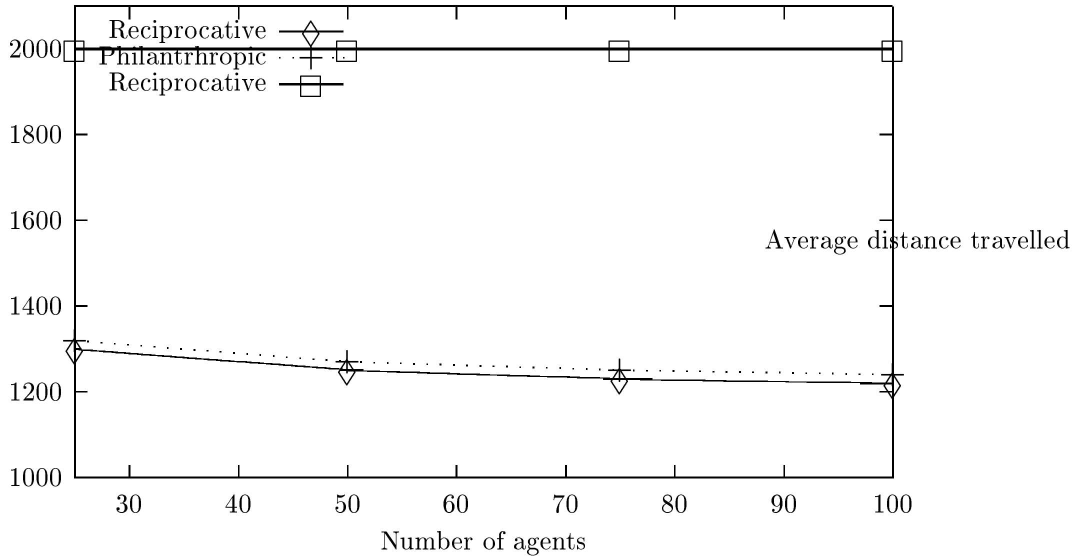
Aufgabenverteilung - Ergebnisse I

Effizienz bei homogenen Agentengruppen:



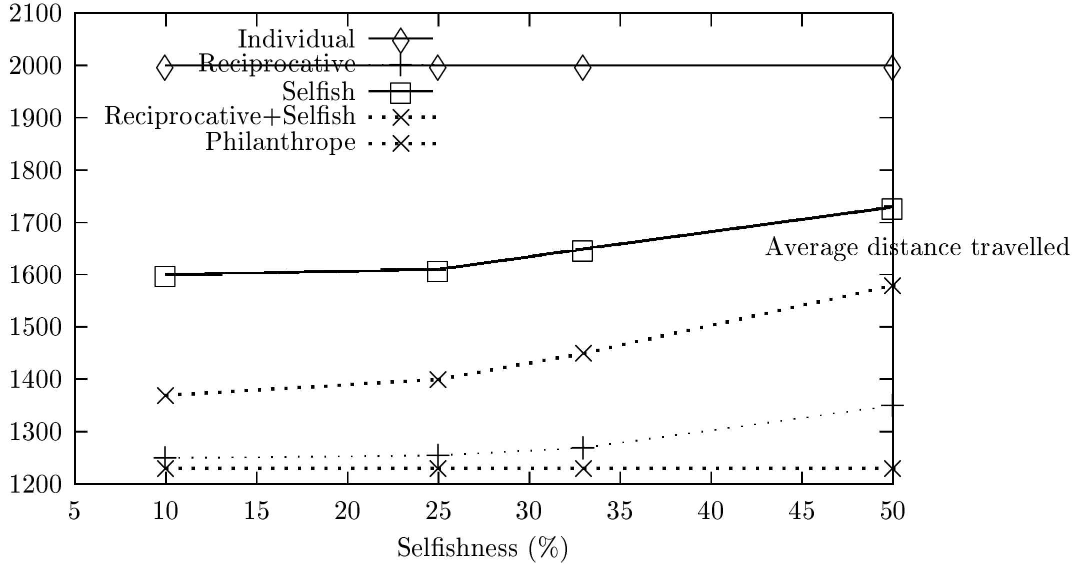
Aufgabenverteilung - Ergebnisse II

Abhängigkeit von der Agentenanzahl:



Aufgabenverteilung - Ergebnisse III

Effizienz bei inhomogenen Gruppen:



Aufgabenverteilung - Ergebnisse IIII

Effizienz bei inhomogenen Gruppen in Abhängigkeit der Paketzustellungen: