# CS 285: Multi-Agent Systems

Fall 2013

Lecture 1

Prof. David C. Parkes

Harvard SEAS

# Lecture 1: Lesson plan

- What is a MAS?
- A retrospective on early MAS research
- Class outline

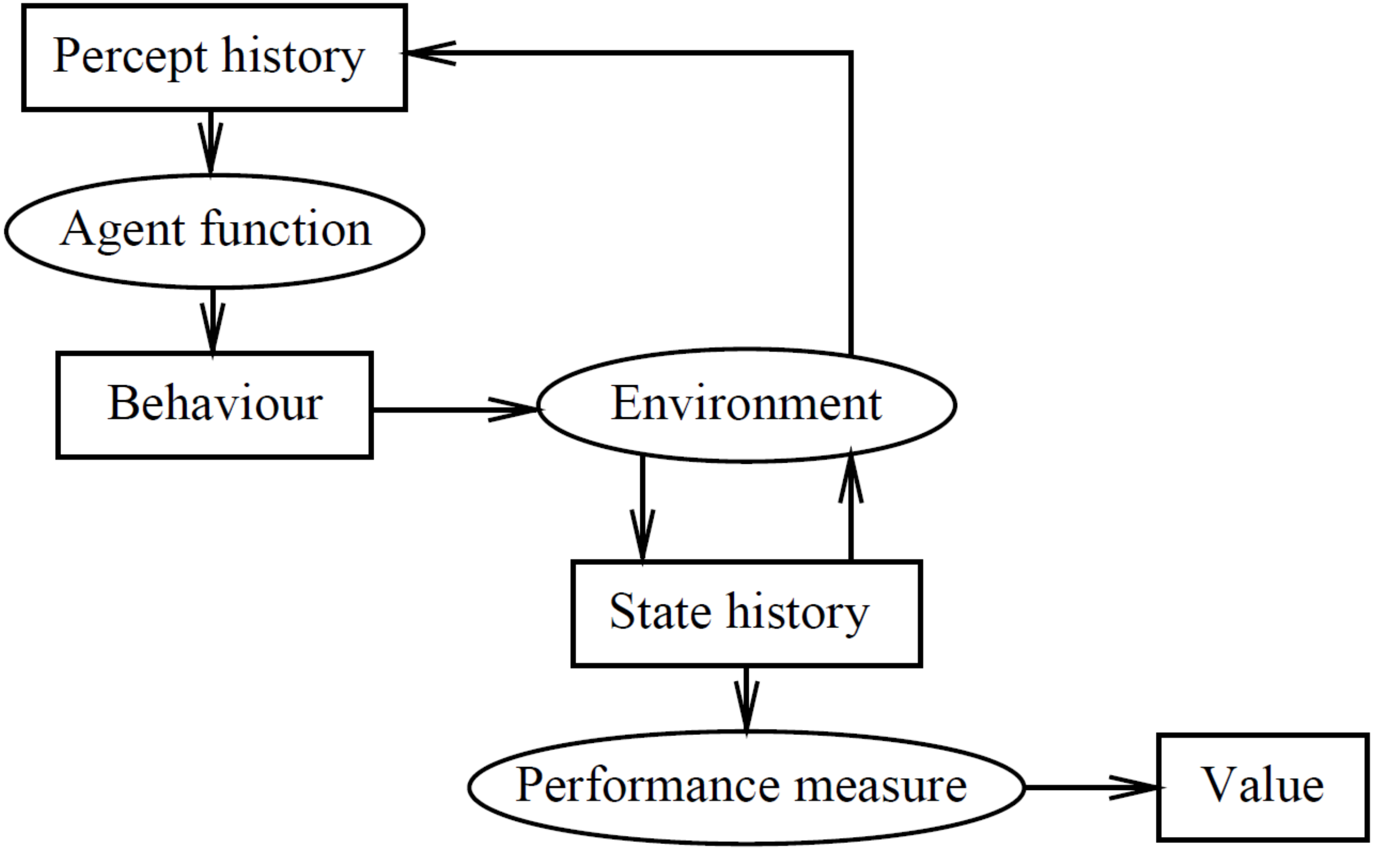
# What is a Multi-Agent System?

- A system with multiple autonomous entities, with distributed information, computational ability, and possibly divergent interests.

- Agents :: artificial or human, cooperative or self-interested

# One view of an agent

(Russell 1997)



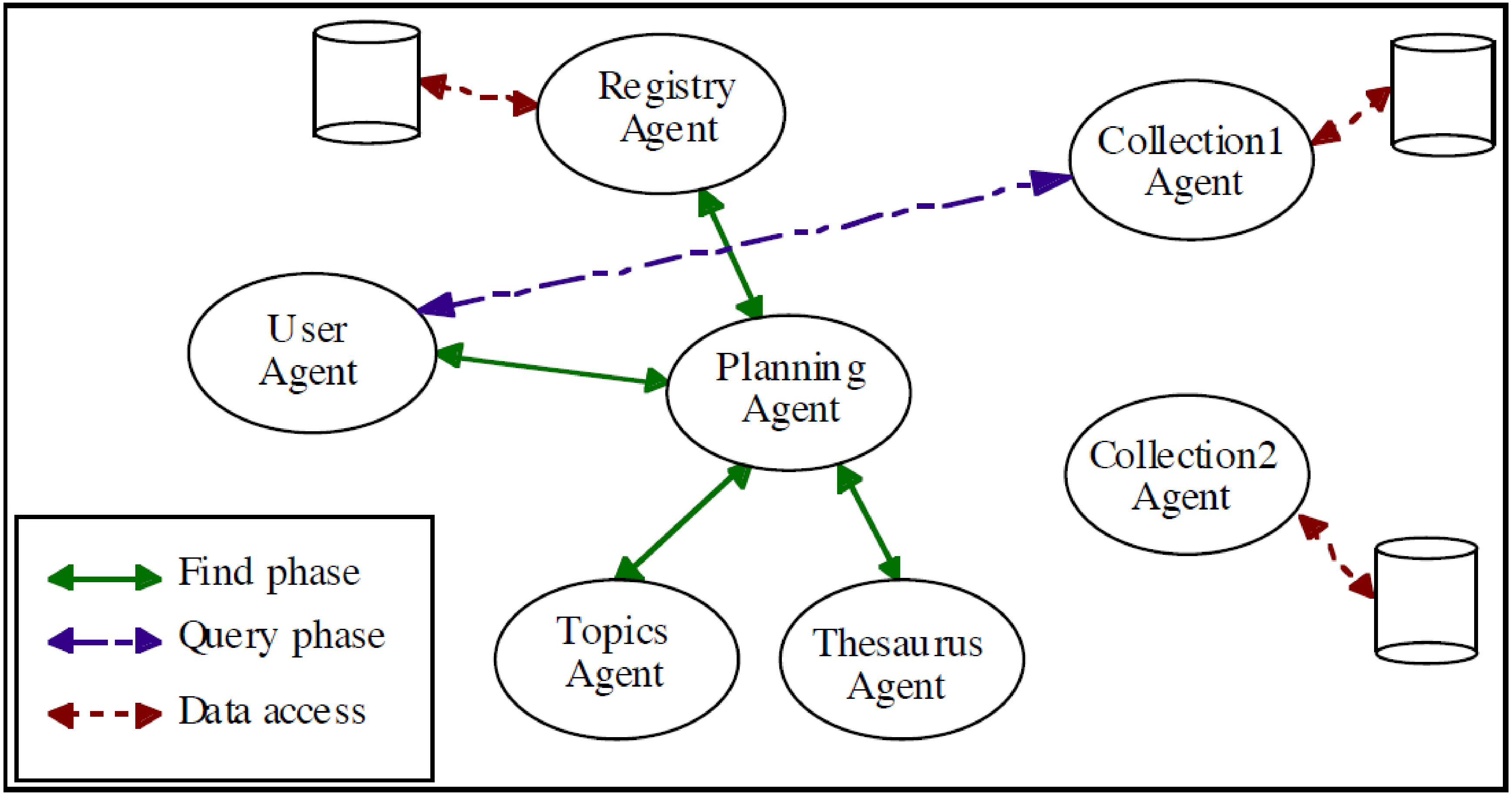
$$l_{opt} = argmax_{l \in \mathcal{L}_M} V(Agent(l, M), \mathbf{E}, U)$$

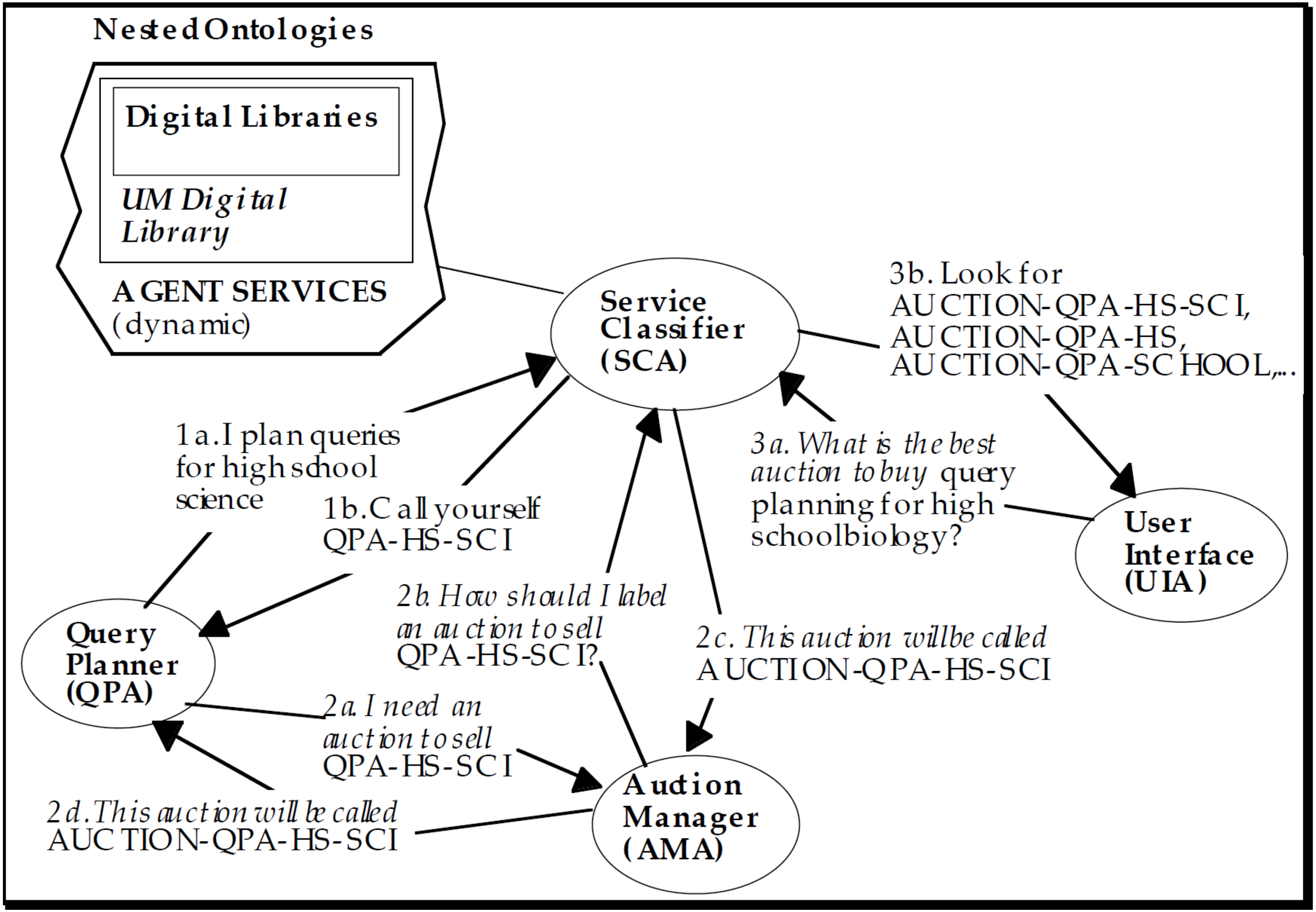
# Two themes of MAS research

- Design of intelligent agents that coordinate or compete with each other
- Design of the coordination environment

# Early example: UM Digital Library

(Weinstein, Birmingham and Durfee 1996-98)



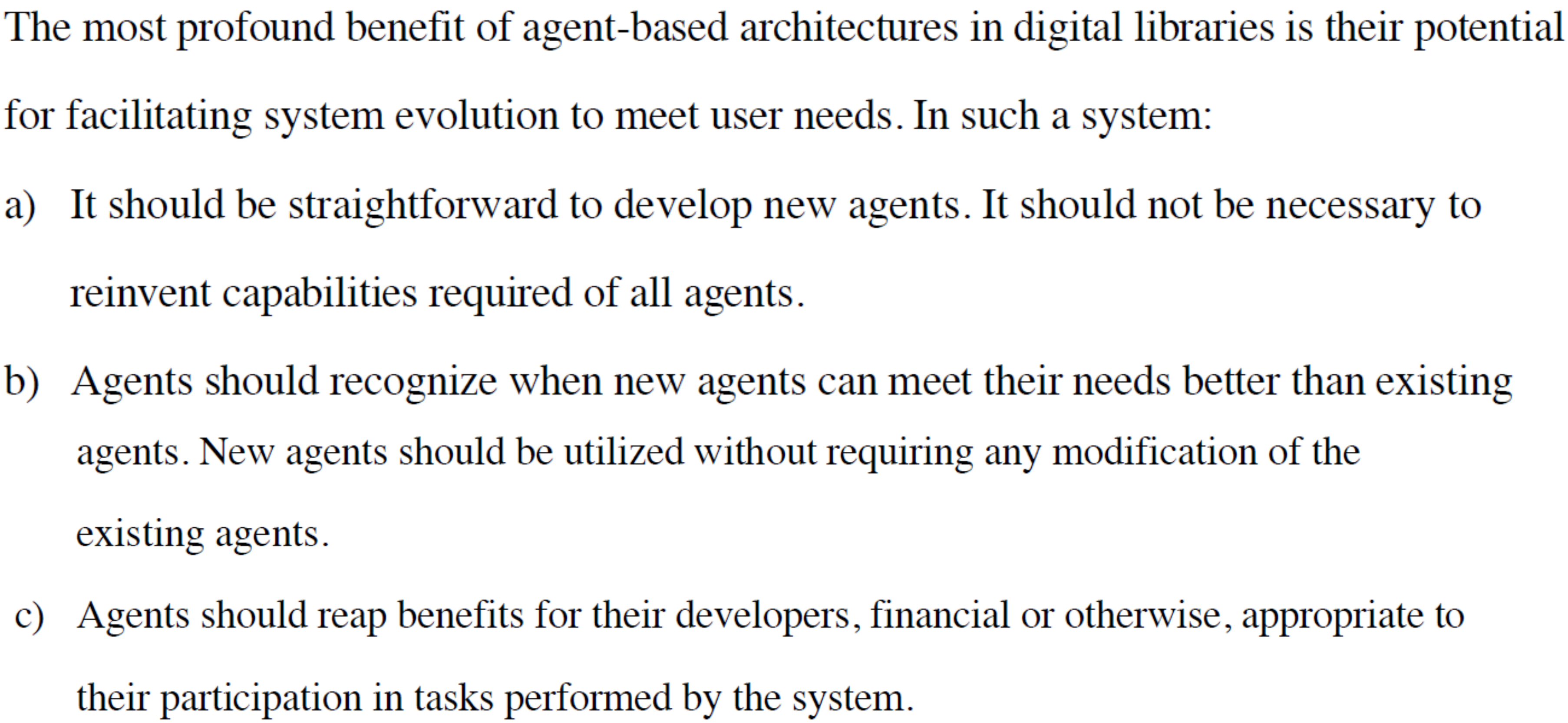

Nested Ontologies
Digital Libraries
UM Digital Library
AGENT SERVICES (dynamic)
Service Classifier (SCA)
3b. Look for AUCTION-QPA-HS-SCI, AUCTION-QPA-HS, AUCTION-QPA-SCHOOL,...
1a. I plan queries for high school science
1b. Call yourself QPA-HS-SCI
3a. What is the best auction to buy query planning for high school biology?
User Interface (UIA)
2b. How should I label an auction to sell QPA-HS-SCI?
2c. This auction will be called AUCTION-QPA-HS-SCI
Query Planner (QPA)
2a. I need an auction to sell QPA-HS-SCI
2d. This auction will be called AUCTION-QPA-HS-SCI
Audion Manager (AMA)

# Agents (as viewed by the UMDL)…

- May team with each other to achieve goals
- Encapsulate well-defined services
- Can make decisions according to prefs.
- May use “mentalistic concepts” such as belief, desire and intention
- Proactive (initiate actions to achieve goals)

# c.f. “agent-oriented programming”

(Shoham; Jennings and Wooldridge)



The most profound benefit of agent-based architectures in digital libraries is their potential for facilitating system evolution to meet user needs. In such a system:

a) It should be straightforward to develop new agents. It should not be necessary to reinvent capabilities required of all agents.

b) Agents should recognize when new agents can meet their needs better than existing agents. New agents should be utilized without requiring any modification of the existing agents.

c) Agents should reap benefits for their developers, financial or otherwise, appropriate to their participation in tasks performed by the system.

# MAS: A Brief History

- ContractNet (Davis and Smith ‘81)
- Consensus (Ephrati and Rosenschein ‘91)
- Distr. CSP (Yokoo et al. ‘92-95; ‘97-05)
- Org. design (Decker and Lesser 93-95)
- Contracts + coalitions (Sandholm &Lesser ‘93-98)
- Market-oriented programming (Wellman ‘93)
- Rules of encounter (Zlotkin and Rosenschein ‘93)
- Multi-agent Inf. Diagrams (Milch and Koller ‘00-01)

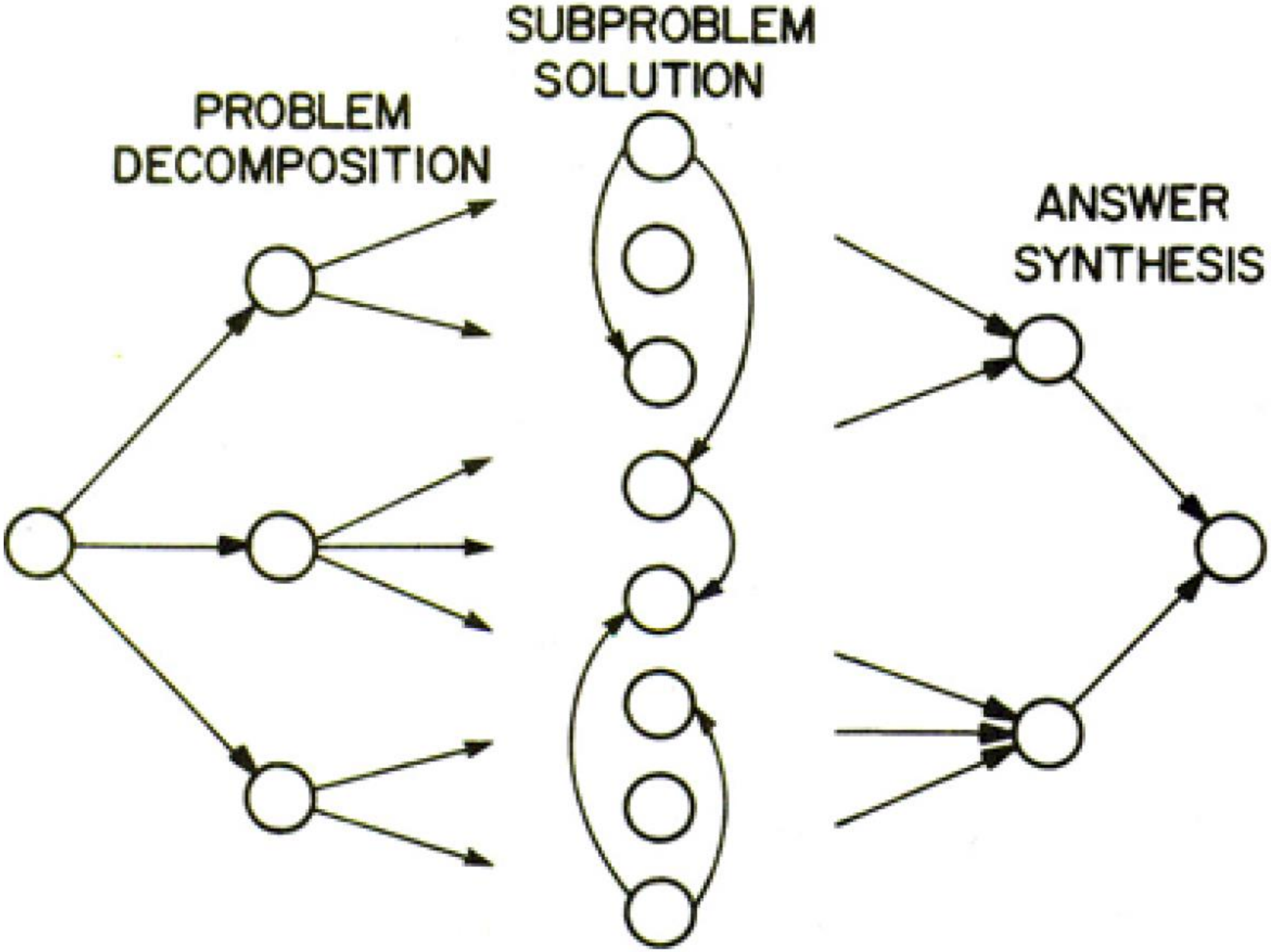
# ContractNet (Smith and Davis ‘81)

# Motivation

- Distributed problem solving
  - – No one has sufficient info to solve entire problem
  - – Control and data distributed
- “How can systems that are perfectly willing to accommodate one another act so as to be an effective team?”
- Nodes (KS’s) cooperate by sharing subtasks of the overall problem

# ContractNet

(Smith and Davis1981)



# “Connection problem”

- Nodes with tasks to execute can find the most appropriate idle nodes to execute them
- Crucial to maintaining the *focus* of the problem solver
- “Most appropriate [agent] to invoke for a task cannot be identified *a priori*”

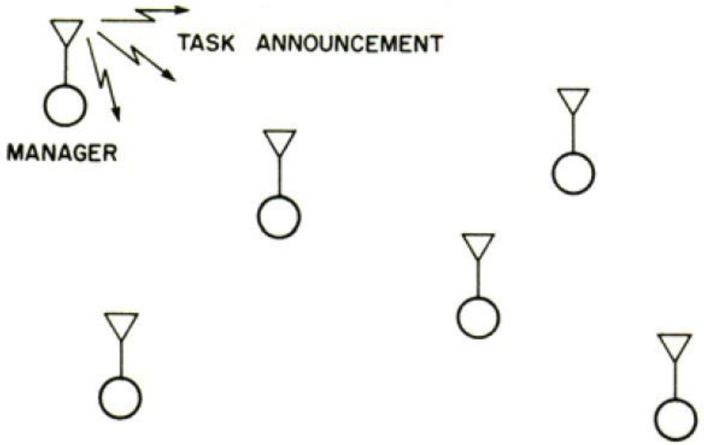


Fig. 3. Sending a task announcement.

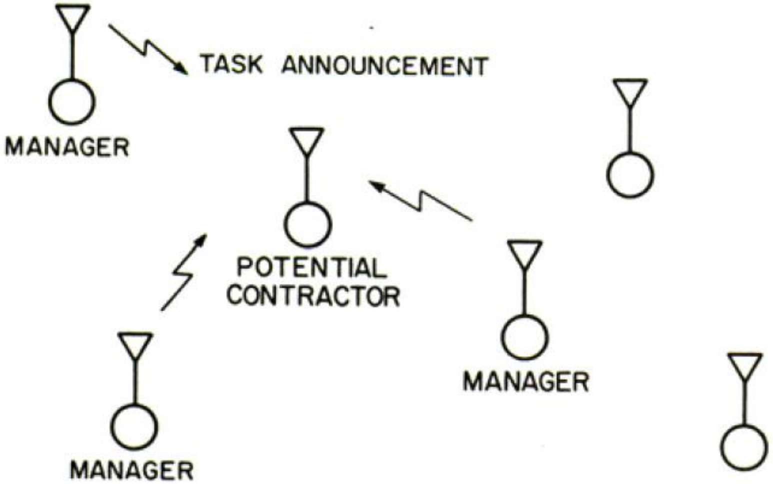


Fig. 4. Receiving task announcements.

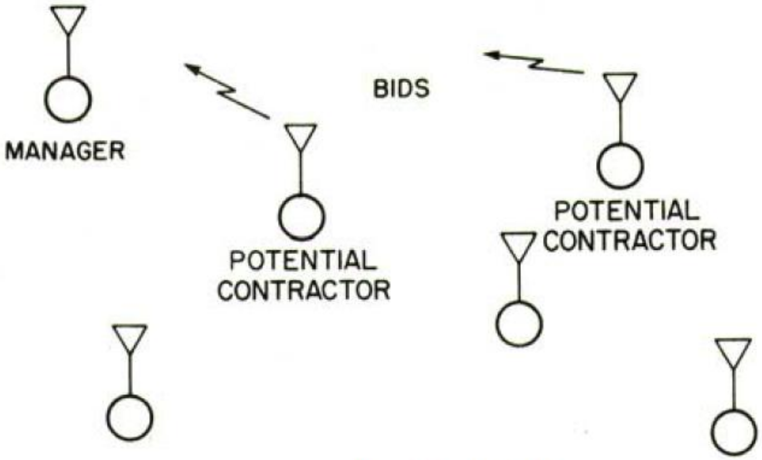


Fig. 5. Bidding.

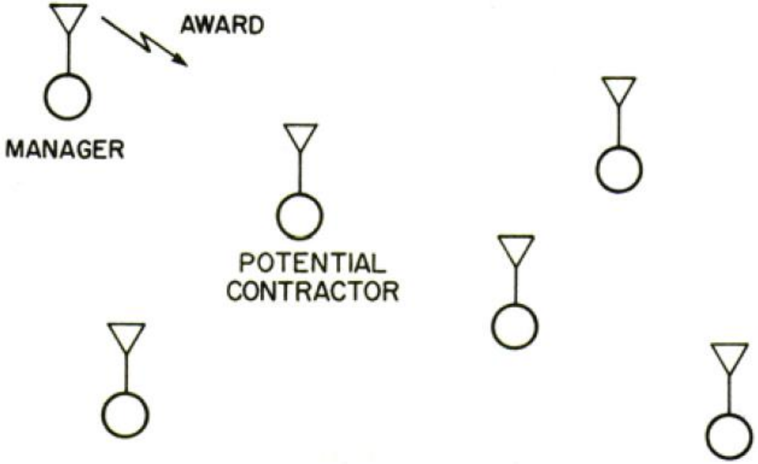


Fig. 6. Making an award.

# ContractNet

- Processors do not get in each other’s way in trying to solve identical subproblems while other subproblems are ignored
- The subproblems that eventually lead to solutions be processed in preference

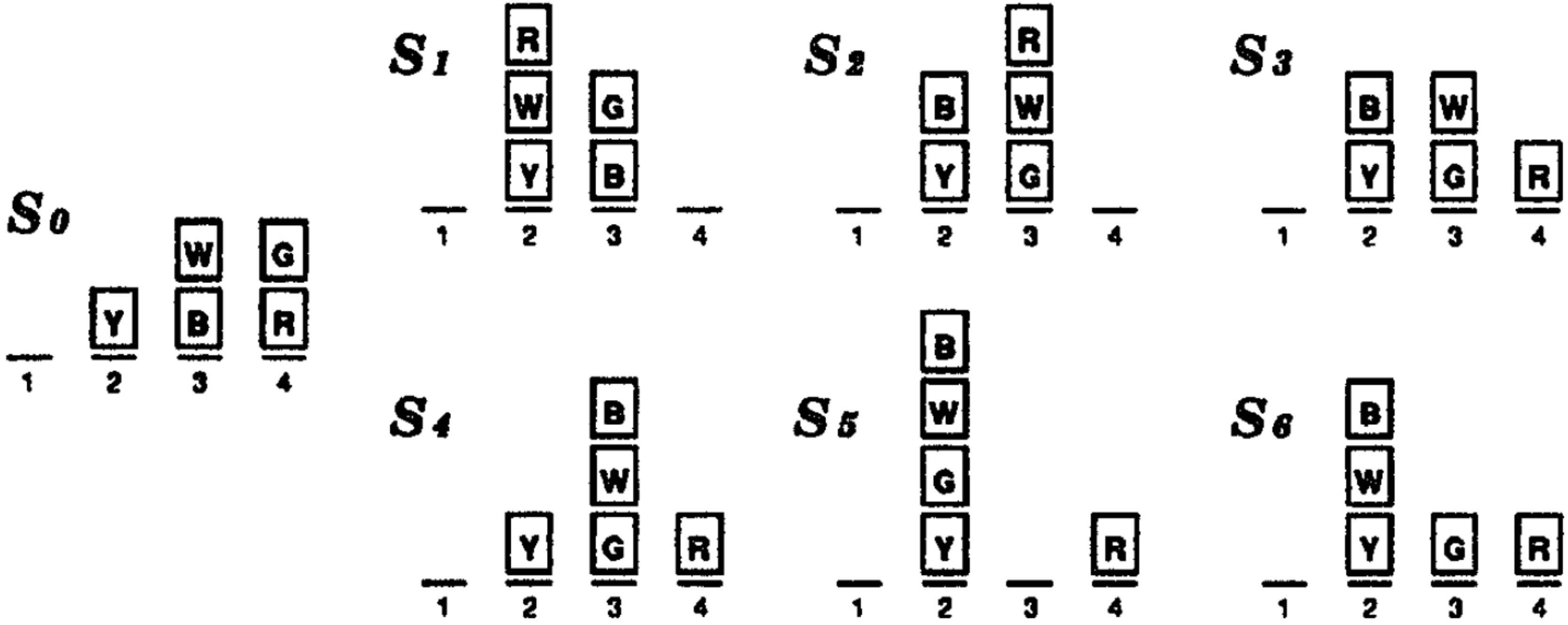
- Specific detail for how to bid not specified…

# Consensus (Ephrati and Rosenschein ‘91)

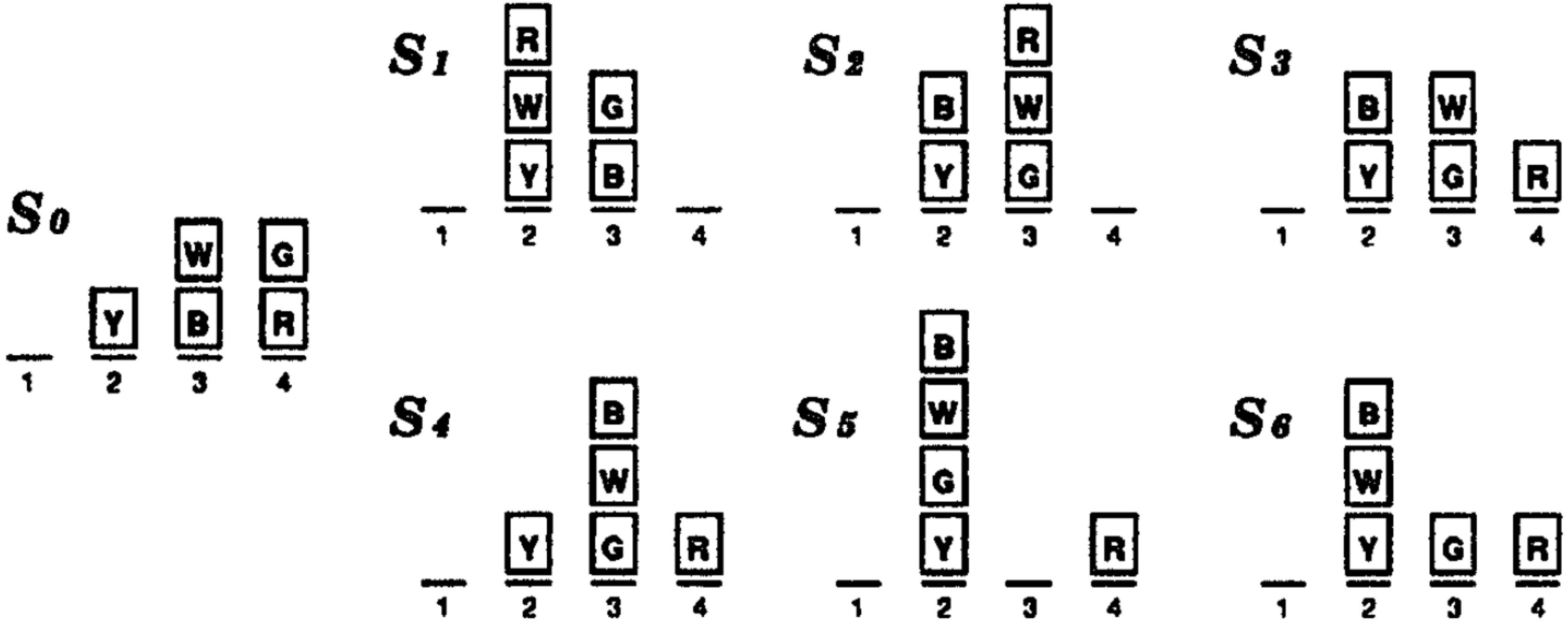
# Motivation

- Autonomous agents need to reach consensus in order to coordinate action
- Bypass negotiation – use a “group choice mechanism” to select the result
- Want one that cannot be manipulated by an untruthful agent

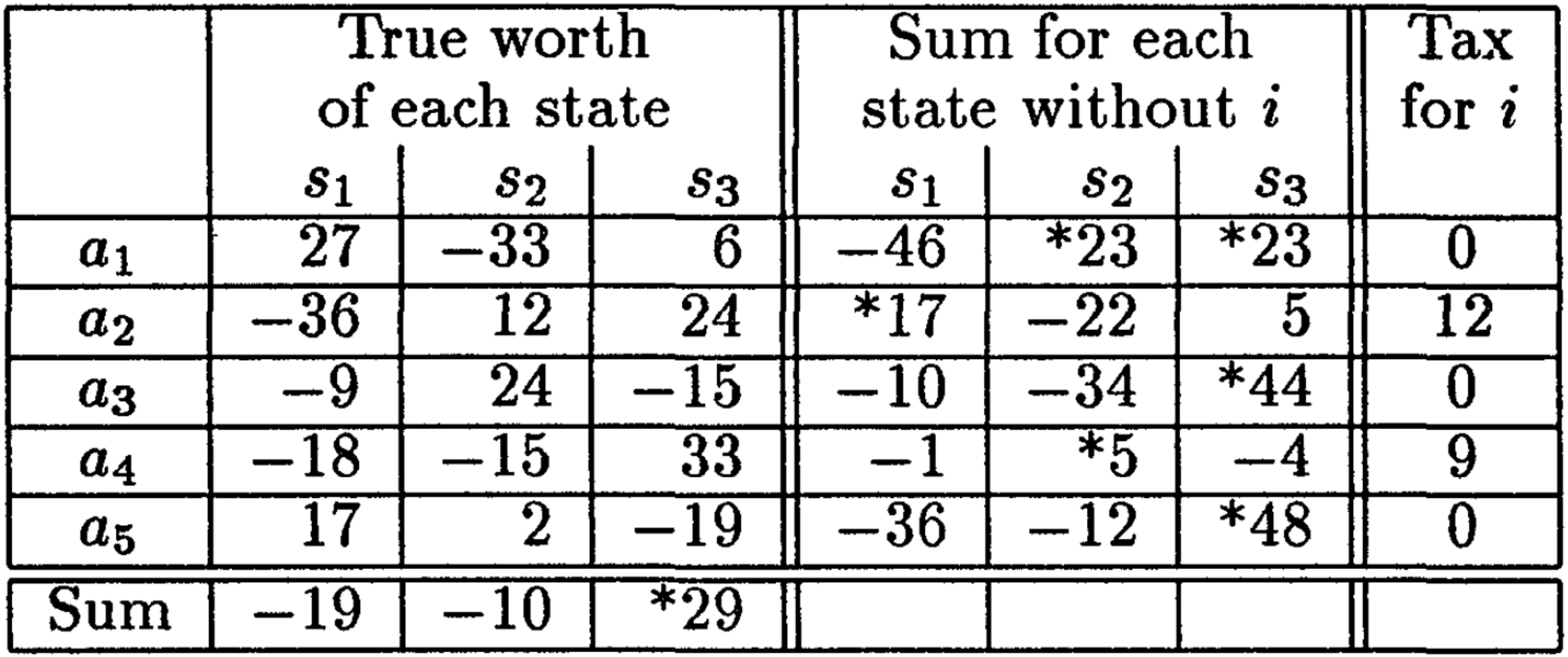
- World in state S0; can move to S1-S6.



- World in state S0; can move to S1-S6.
- Goals; g_1{At(G,3), At(W,2)}; g_2= {On(W,G), On(R,W)}
- v_i(S) = cost_i[reach goal, S0] – cost_i[reach goal, S]; e.g., v_1 = (2, 0, 1, 0, -2, 2)

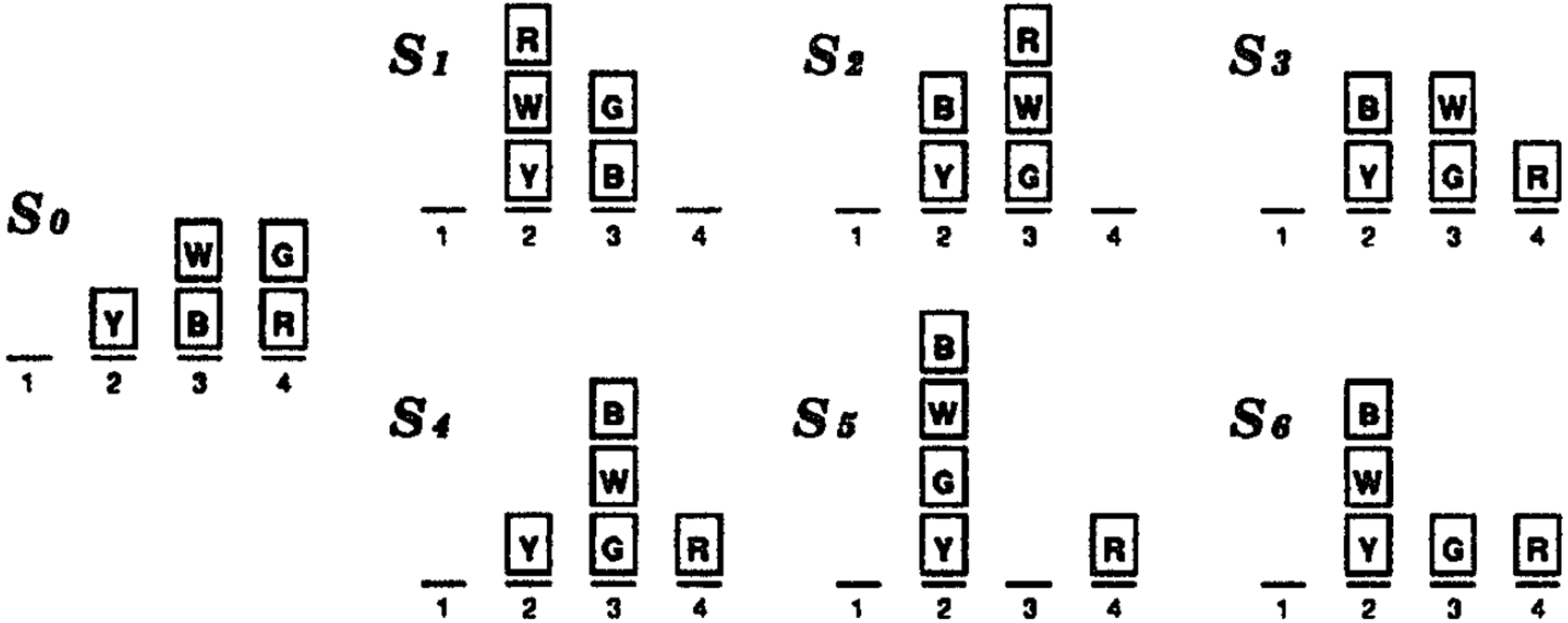


# Clarke tax – *collect bids and fine a tax equal to the portion of bid that made a difference*



| | True worth of each state | | | Sum for each state without $i$ | | | Tax for $i$ |
|---|---|---|---|---|---|---|---|
| | $s_1$ | $s_2$ | $s_3$ | $s_1$ | $s_2$ | $s_3$ | |
| $a_1$ | 27 | −33 | 6 | −46 | *23 | *23 | 0 |
| $a_2$ | −36 | 12 | 24 | *17 | −22 | 5 | 12 |
| $a_3$ | −9 | 24 | −15 | −10 | −34 | *44 | 0 |
| $a_4$ | −18 | −15 | 33 | −1 | *5 | −4 | 9 |
| $a_5$ | 17 | 2 | −19 | −36 | −12 | *48 | 0 |
| Sum | −19 | −10 | *29 | | | | |

Figure 1: Calculating the Clarke Tax



$\{At(G,3), At(W,2)\}$, $g_2 = \{On(W,G), On(R,W)\}$,
$g_3 = \{On(B,W), At(W,3)\}$. Assume that each *Move*

$\langle 2,0,1,0,-2,2\rangle, \langle 0,3,2,1,1,0\rangle, \langle -1,2,3,4,1,1\rangle$.
$s_3$ (which is only one *Move* operation distant from *all*
the agents' goals) is chosen, and no tax is collected.

# Discussion

- How to generate alternatives
- Different ways to determine “worth”
- Handling tax waste
- Work distribution

# Distributed CSP
# (Yokoo et al. ‘92-95; ‘97-05)

# Distr. Constraint Satisfaction

(Yokoo, Durfee, Ishida, Kuwabara 1992-95)

- n variables x_1,… x_n
- Finite domains D_1,…, D_n
- Each agent belongs to one agent
- Constraint predicates p_k(x_1,..x_m) distributed amongst agents

- Goal: assign values to variables so that all predicates satisfied

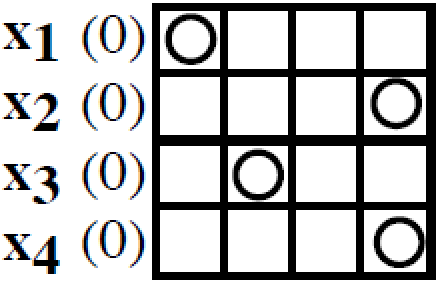
# DCSP :: Motivation

- Coordination of artificial automated agents; “Important infrastructure in DAI”

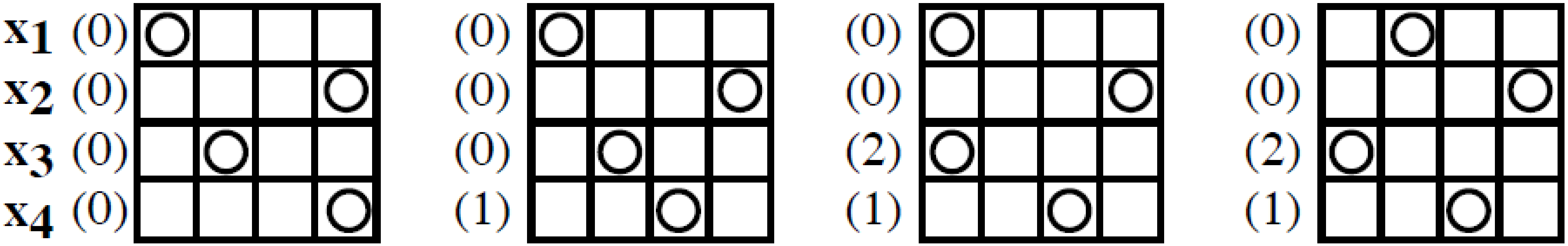
Examples:

- Distributed truth maintenance
  - assign “IN” or “OUT” to data, some data shared
- Resource allocation
  - assign plans to the task(s) of each agent s.t. all plans can be executed simultaneously

# Toy example: n-Queens



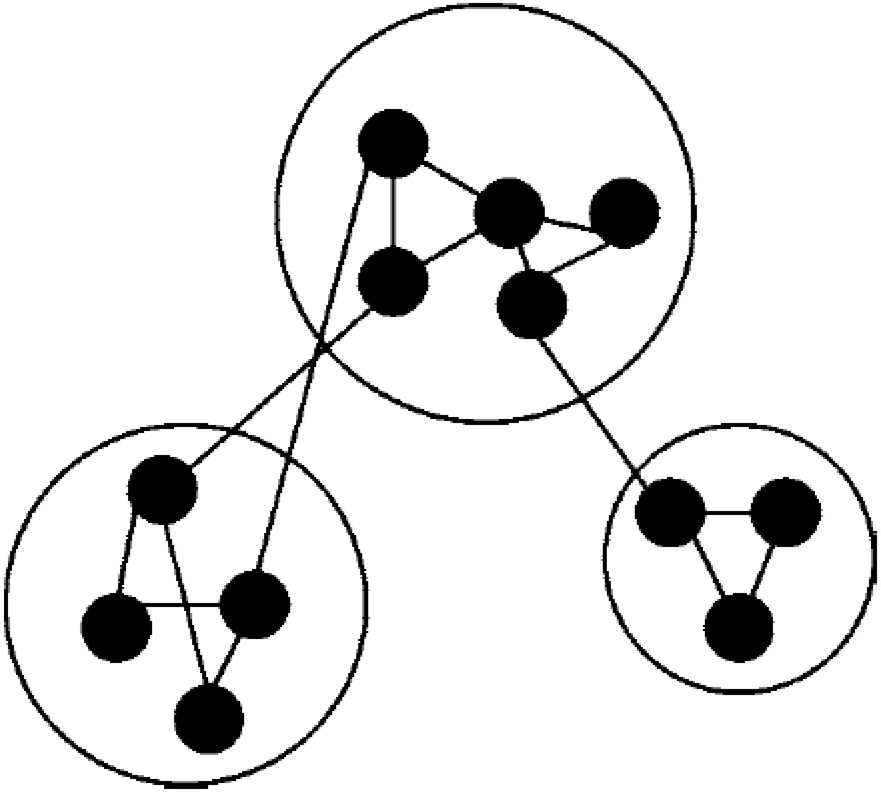
# Toy example: n-Queens



- Asynchronous Weak commitment
  - – Assign, send messages, if in conflict then try to fix (reduce constraints) and increment priority
  - – Priority by agent ID if priority numbers the same

# Extension: Optimization!

(Yokoo et al. 1997- ;  Shen, Tambe, Yokoo, 2003-05)



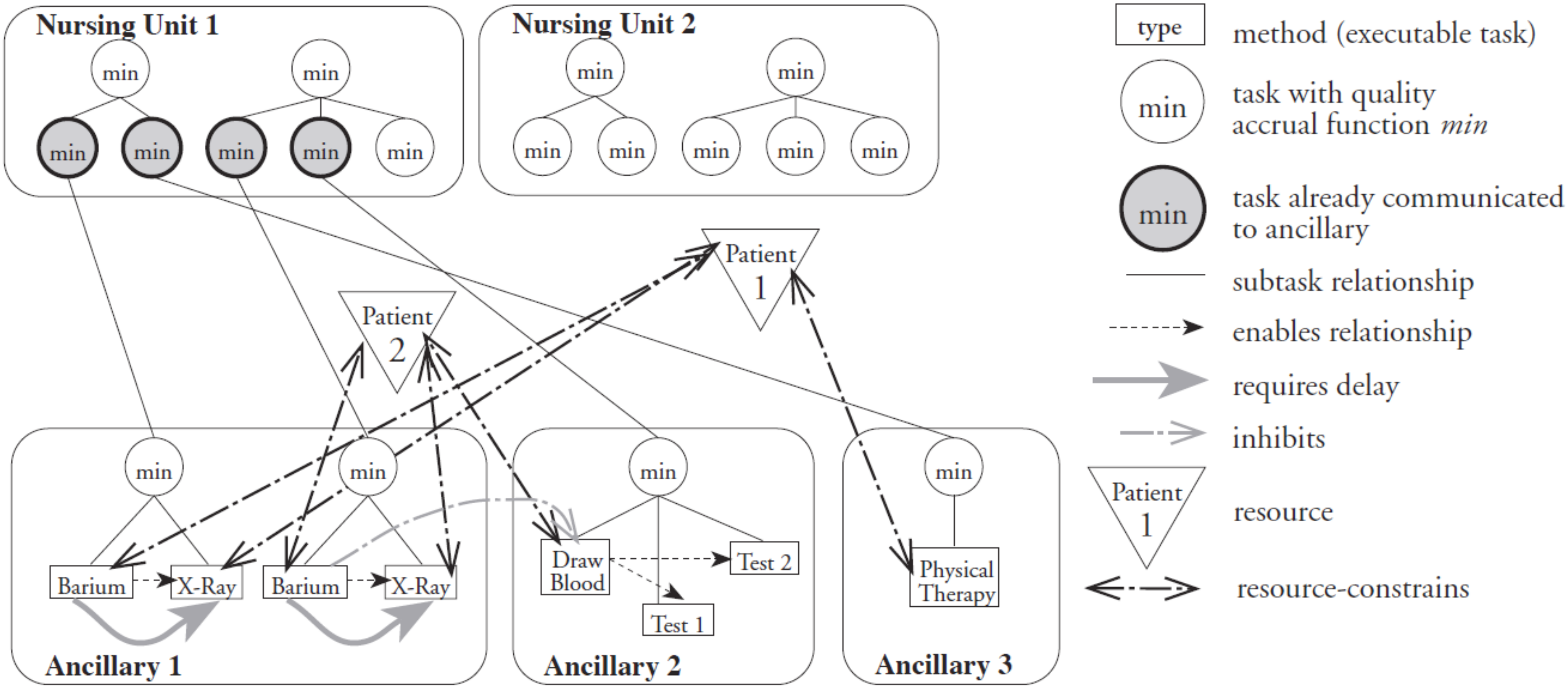
$$F(\mathcal{A}) = \sum_{x_i, x_j \in V} f_{ij}(d_i, d_j), \quad \text{where } x_i \leftarrow d_i$$

# Organization Design for Task Oriented Environments (Decker and Lesser 93-95)

# TAEMS :: Motivation

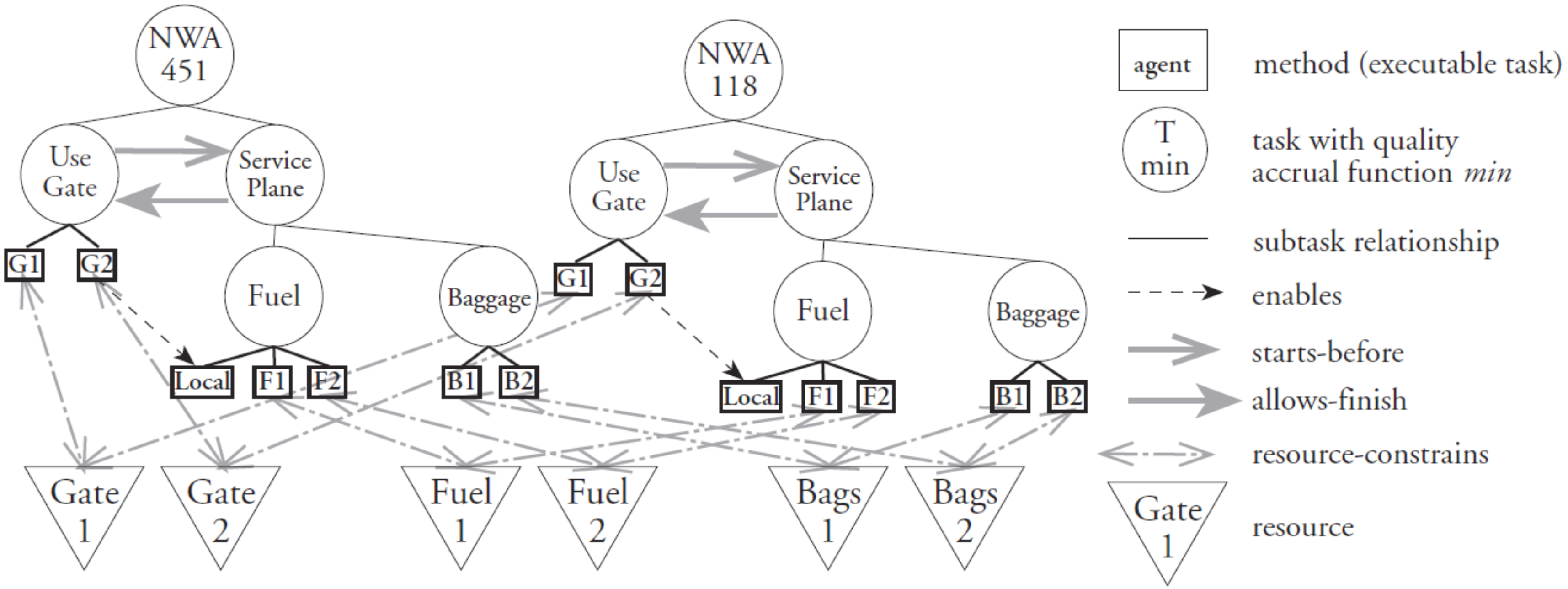
- Organizational-based framework for representing coordination problems in a formal, domain-independent way
- Tool for building and testing computational theories of coordination
    - Task interrelationships (hard – enables, soft – facilitates)
    - Task group, task (set of subtasks), executable method

# Example: Hospital scheduling



Units – scheduling agents minimize patients' stays
Ancillary agents – maximize equipment use, minimize setup times

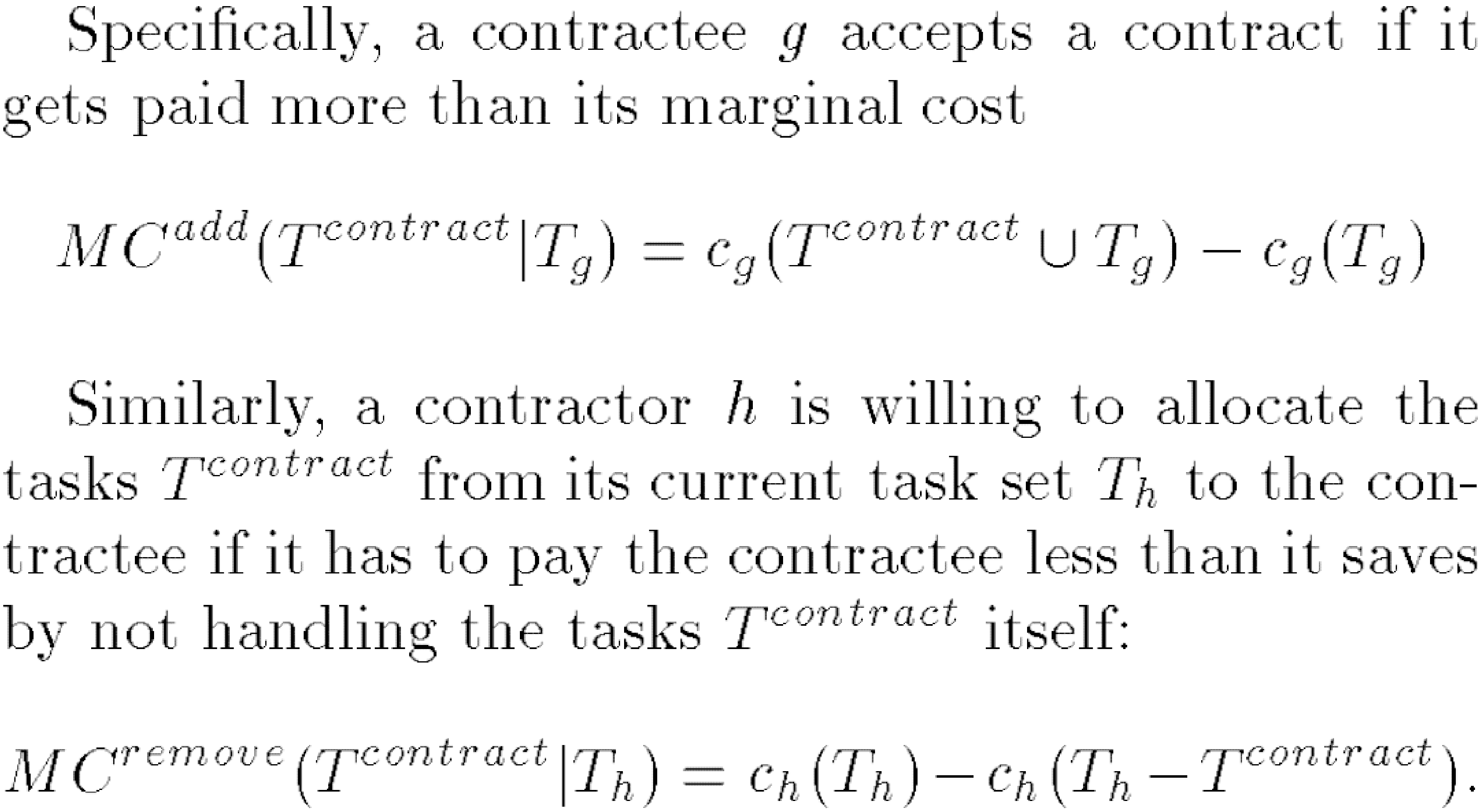
# Example: Airport scheduling



# Task reallocation (Sandholm and Lesser ‘93-98)

# Marginal-cost Based Contracting

(Sandholm and Lesser 1993-98)

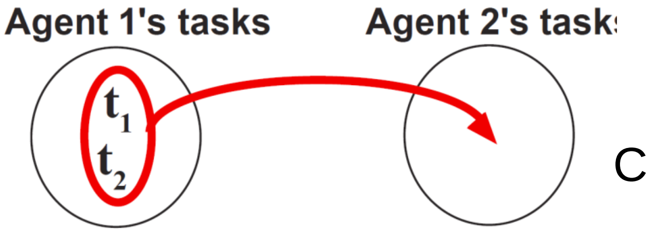


Specifically, a contractee $g$ accepts a contract if it gets paid more than its marginal cost

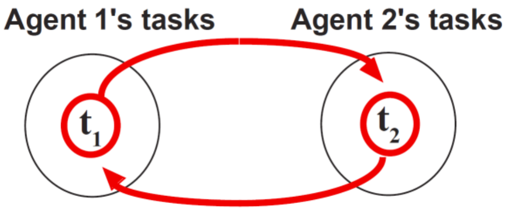
$$MC^{add}(T^{contract}|T_g) = c_g(T^{contract} \cup T_g) - c_g(T_g)$$

Similarly, a contractor $h$ is willing to allocate the tasks $T^{contract}$ from its current task set $T_h$ to the contractee if it has to pay the contractee less than it saves by not handling the tasks $T^{contract}$ itself:

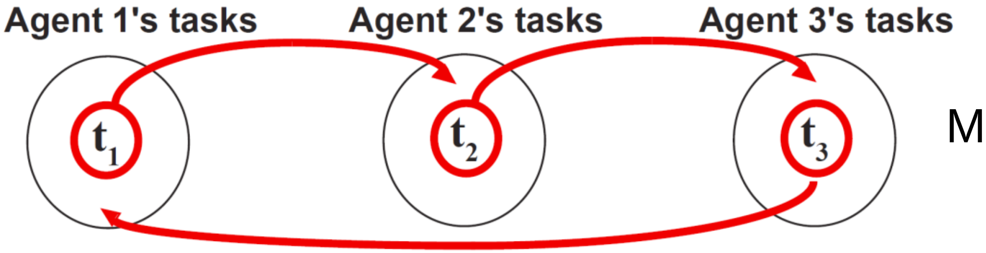
$$MC^{remove}(T^{contract}|T_h) = c_h(T_h) - c_h(T_h - T^{contract}).$$



Cluster contract

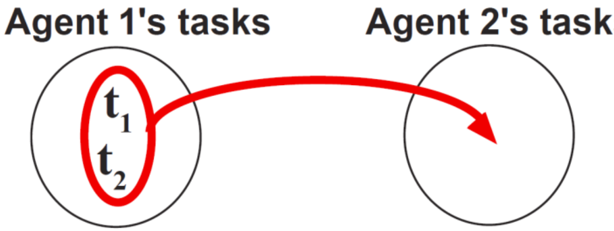


Swap contract

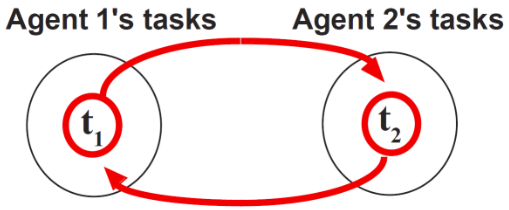


Multi-agent contract

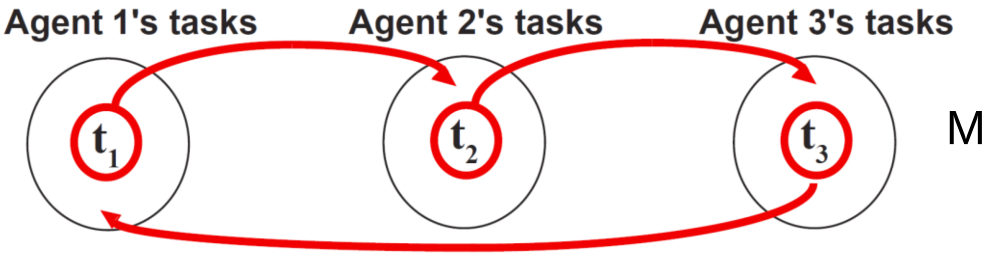
Find “IR” paths that (a) avoid local suboptimality, (b) have “anytime” property and avoid need to backtrack



Cluster contract



Swap contract



Multi-agent contract

Find “IR” paths that (a) avoid local suboptimality, (b) have “anytime” property and avoid need to backtrack

Claim: even M contracts insufficient..

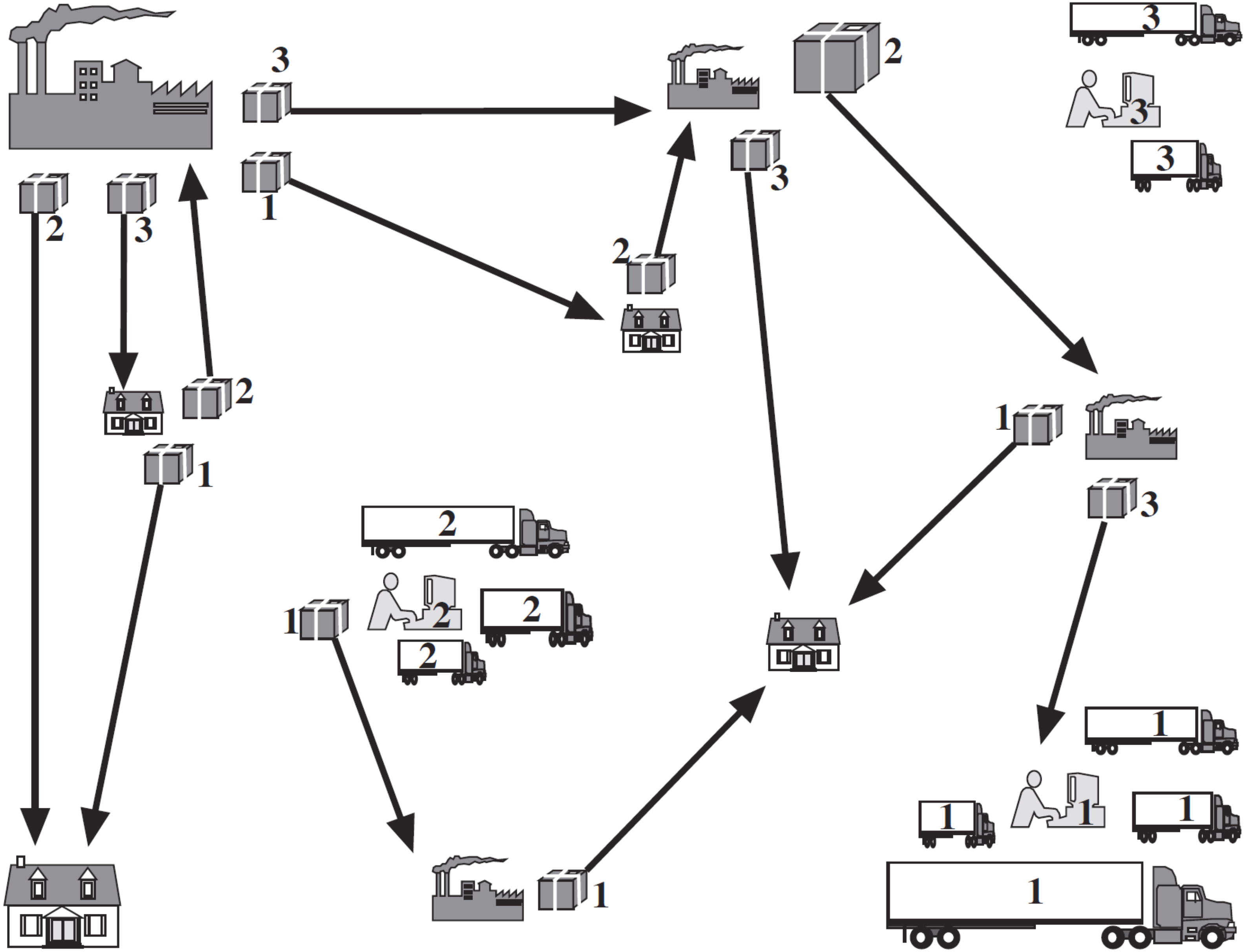
agent 1 (H): Task

agent 2 (L): No task

# Dynamic Coalition Formation

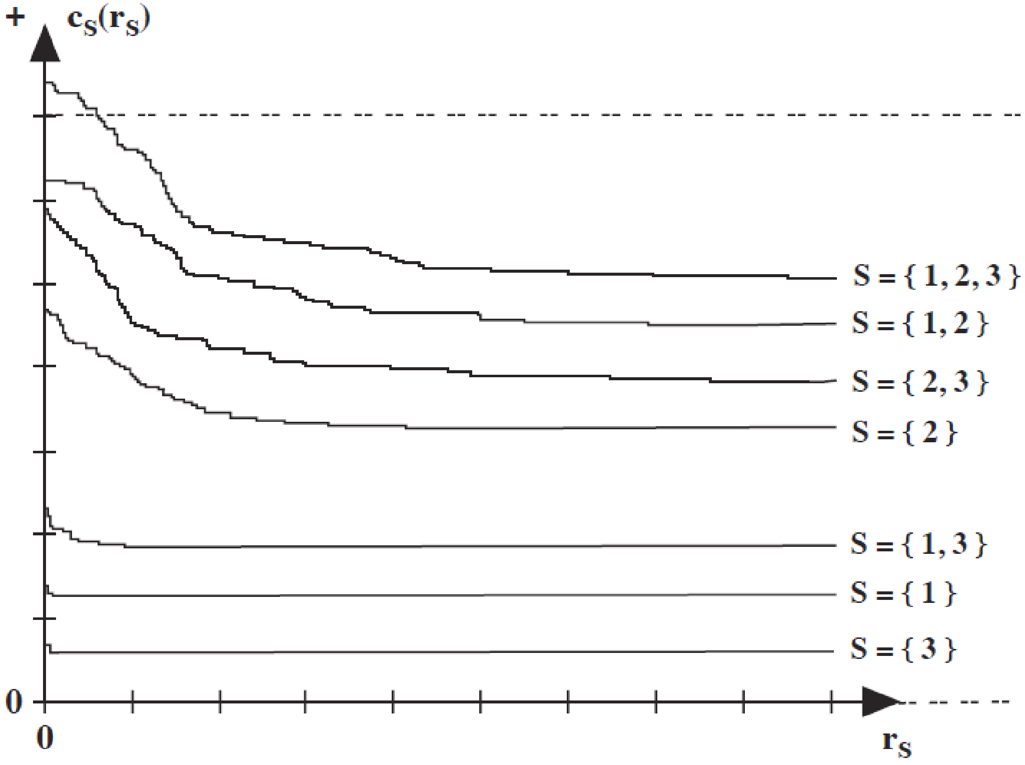
(Sandholm and Lesser 1995)

- Motivations
- “small transaction commerce on the Internet”
- “industrial trend towards dynamic, virtual enterprises that can take advantage of economies of scale”


3
2
1
2
3
2
1
3
2
3
2
1
3
3
3
2
2
2
1
1
1
1
1
1
1

- Three interrelated challenges:
  - – Generate coalitions
  - – Solve the optimization problem for each coalition
  - – Divide the value of generated solution

*anytime algs. for solving optimization problem for a coalition*



# Market-oriented programming (Wellman ‘93)

# Market-oriented programming

(Wellman 1993)

Consumer:

$$\max_{\mathbf{x}_i} u_i(\mathbf{x}_i) \text{ s.t. } \mathbf{p} \cdot \mathbf{x}_i \leq \mathbf{p} \cdot \mathbf{e}_i$$

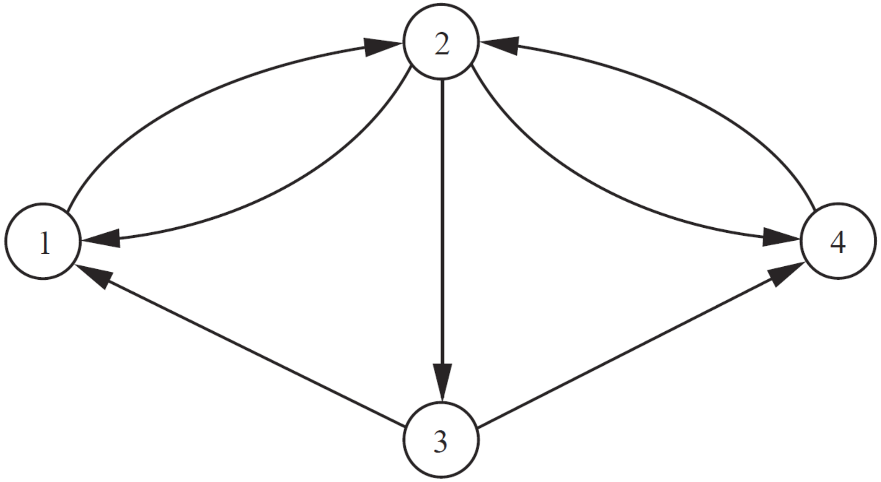
Producer:

$$\min_{\mathbf{x}_i} \mathbf{p} \cdot \mathbf{x}_i \text{ s.t. } -x_{i,j} \leq f_i(\mathbf{x}_{i,\bar{j}})$$

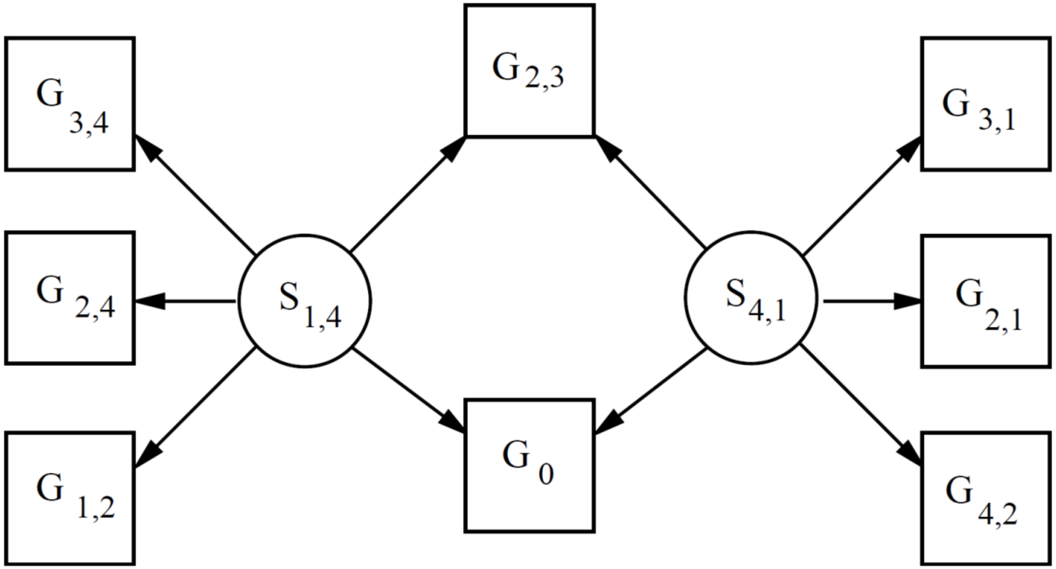
Competitive equilibrium : agents best respond, and total consumption = total production

… WALRAS tatonnement algorithm

Example: Transportation Network

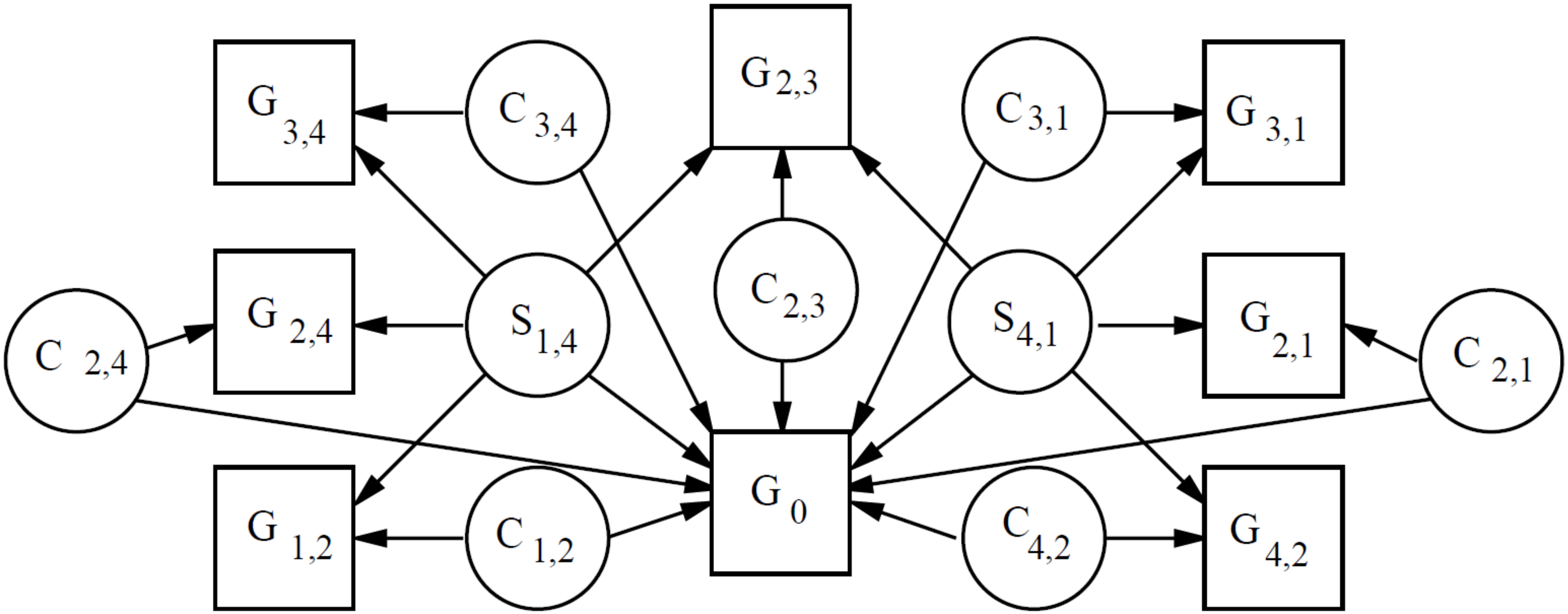


… the economy



Sub-optimality: over-use of (2,3)

# Introducing “carriers” (producers):



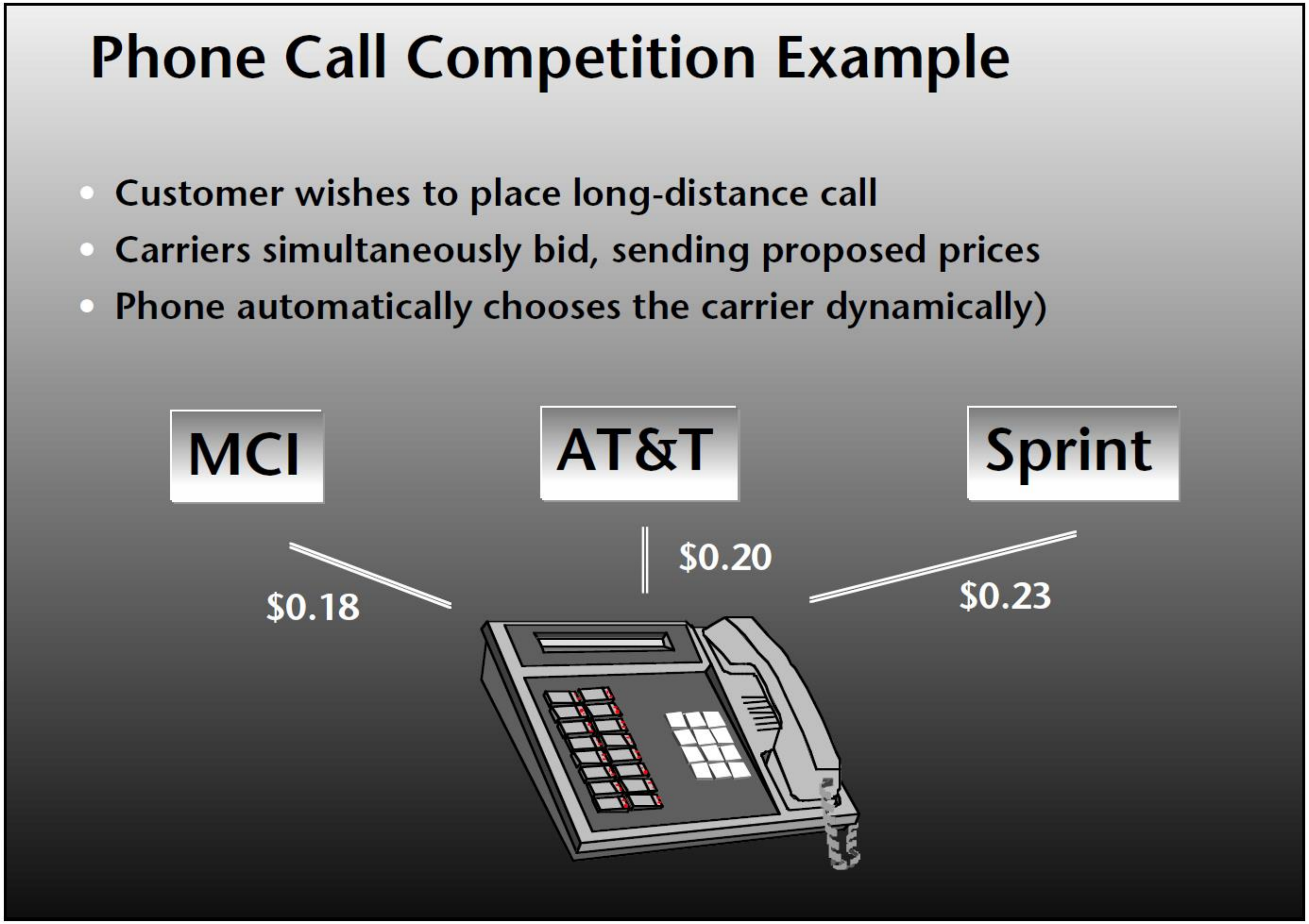
… set price on goods at marginal cost

| pricing | TC | expense | profit | $p_{1,2}$ | $p_{2,1}$ | $p_{2,3}$ | $p_{2,4}$ | $p_{3,1}$ | $p_{3,4}$ | $p_{4,2}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| MC (SE) | 1136 | 1514 | 378 | 40.0 | 35.7 | 22.1 | 35.7 | 13.6 | 13.6 | 40.0 |
| AC (UE) | 1143 | 1143 | 0 | 30.0 | 27.1 | 16.3 | 27.1 | 10.7 | 10.7 | 30.0 |

# Rules of Encounter (Zlotkin and Rosenschein ‘93)

# Rules of Encounter

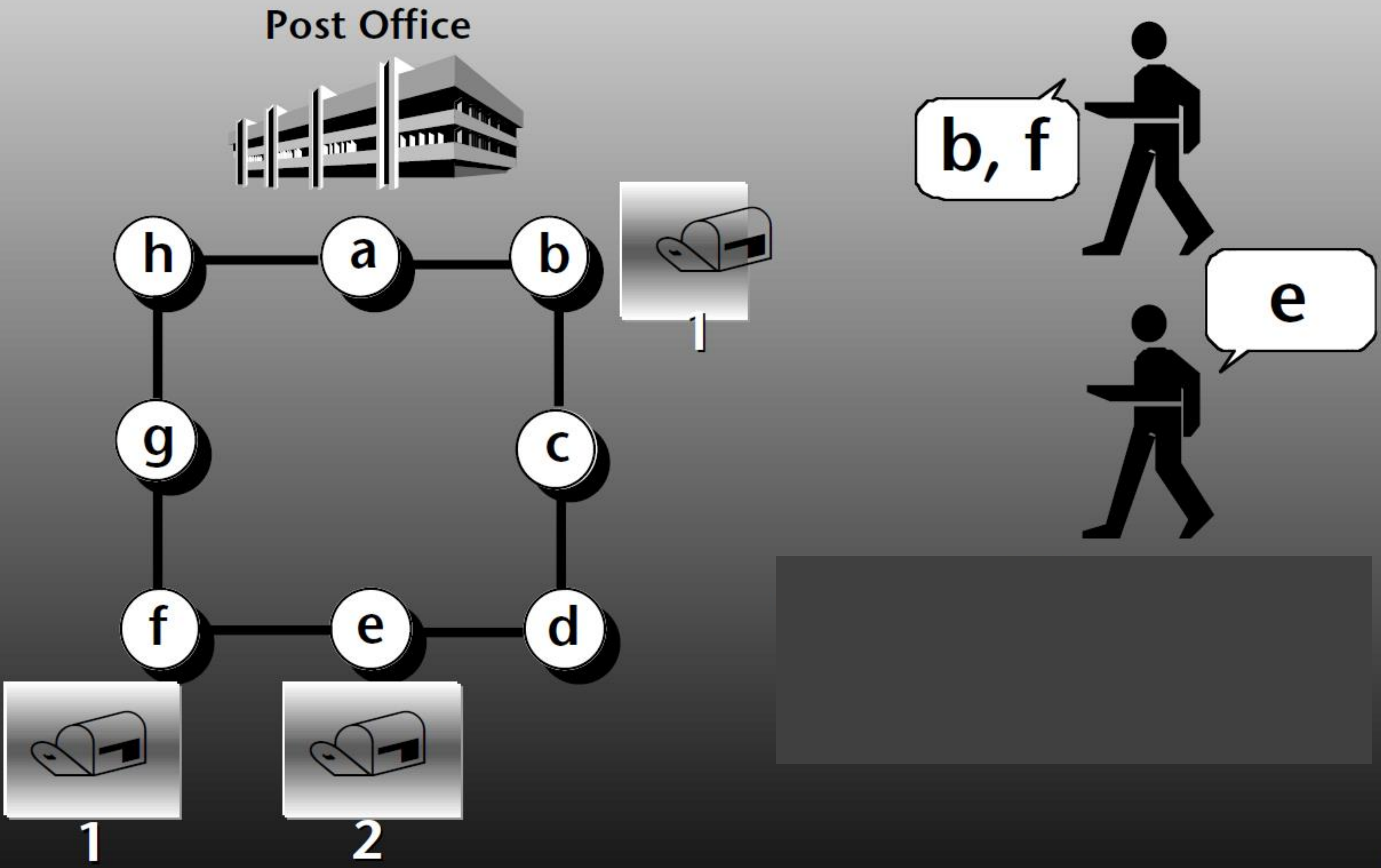
(Rosenschein and Zlotkin 1993-94)



## Phone Call Competition Example

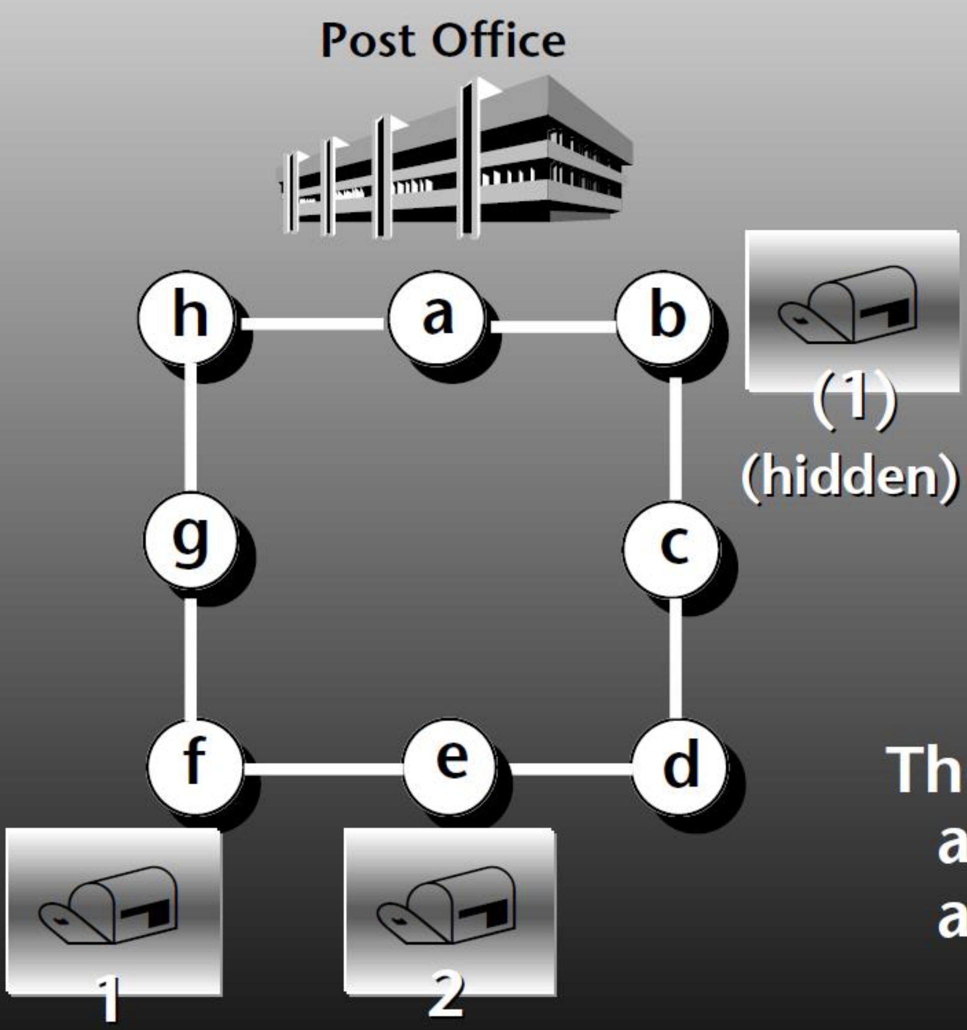
- Customer wishes to place long-distance call
- Carriers simultaneously bid, sending proposed prices
- Phone automatically chooses the carrier dynamically)

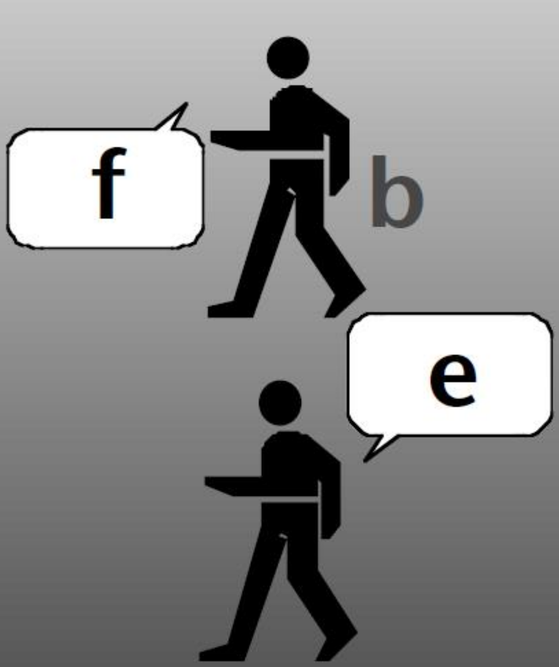
# –1 Phase Game: Broadcast Tasks



a

# Hiding Letters

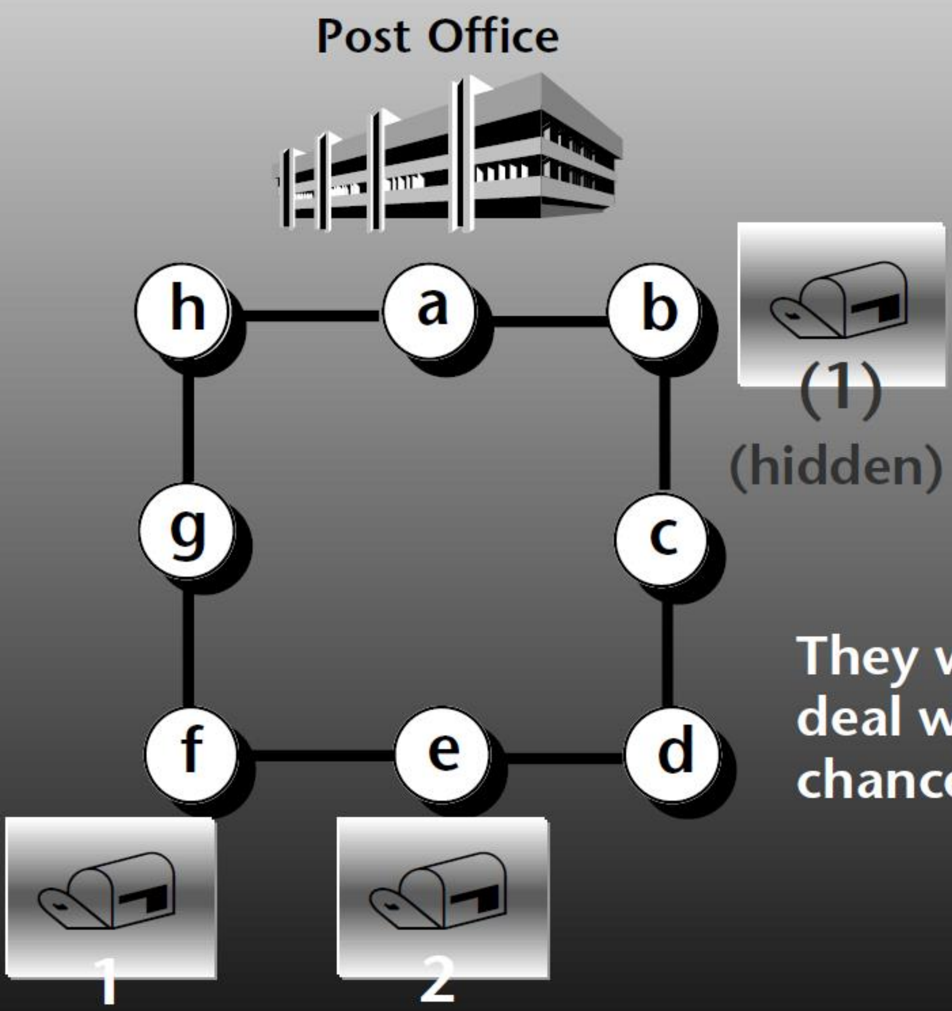


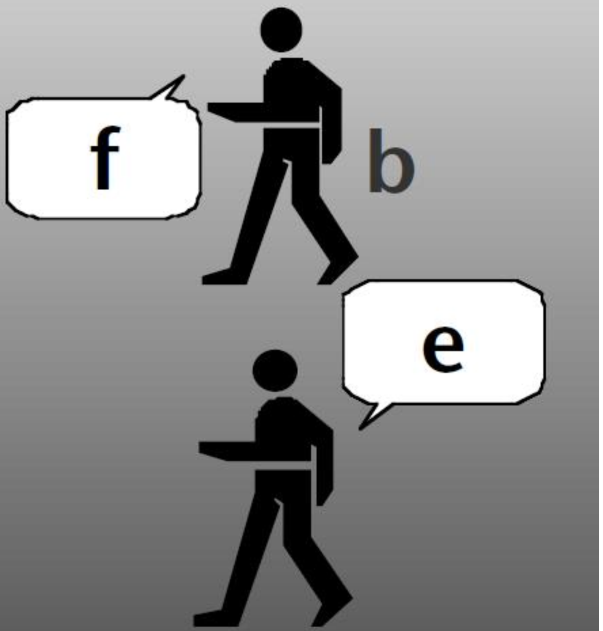


They then agree that agent 2 delivers to f and e.

b

# Hiding Letters with Mixed All-or-Nothing Deals

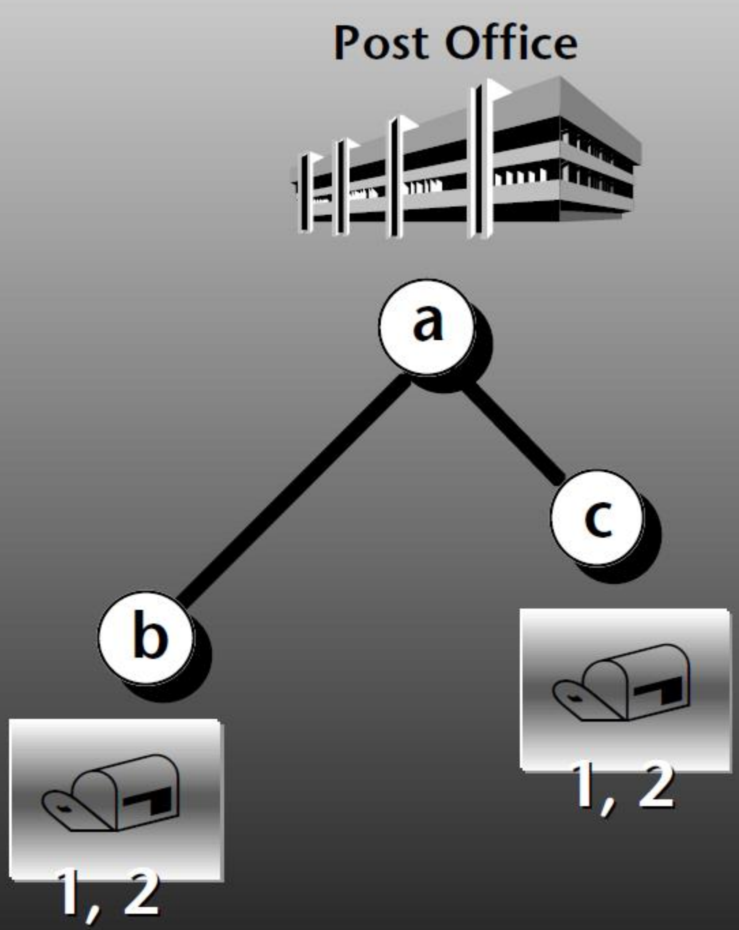


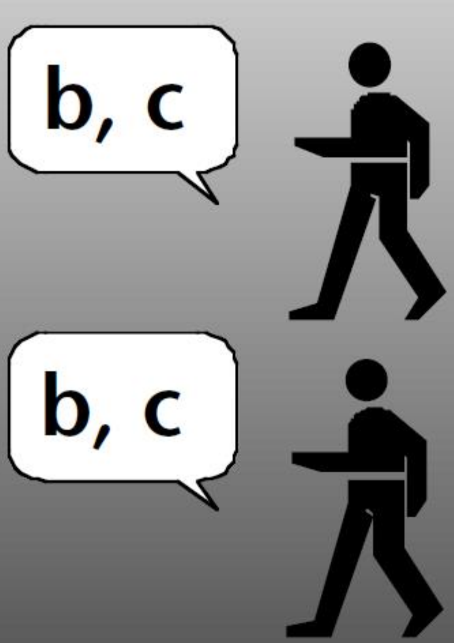


They will agree on the mixed deal where agent 1 has a 3/8 chance of delivering to f and e.

a

# Another Possibility for Deception

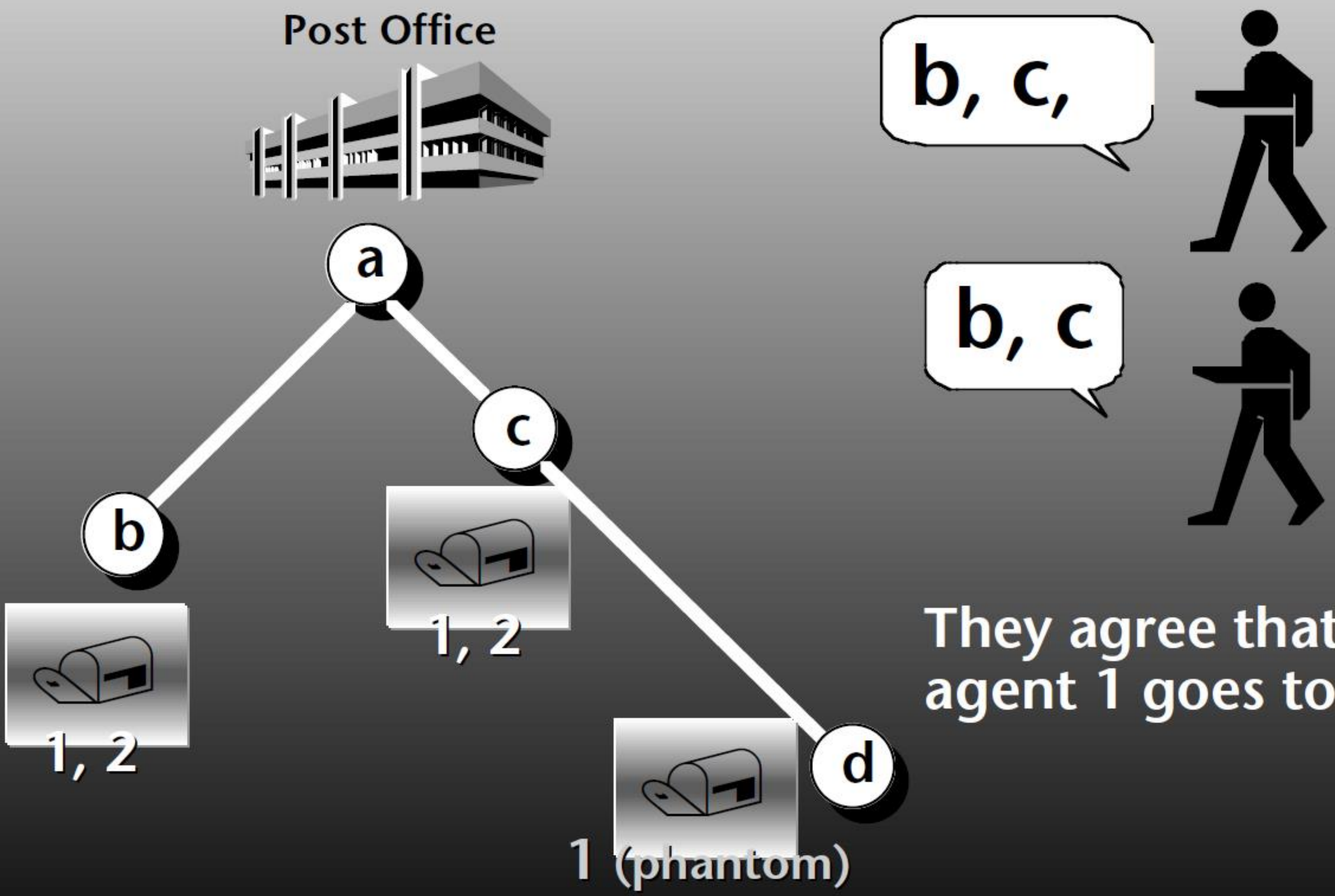




They will agree to flip a coin to decide who goes to b and who goes to c.

a

# Phantom Letter



They agree that agent 1 goes to c.

b

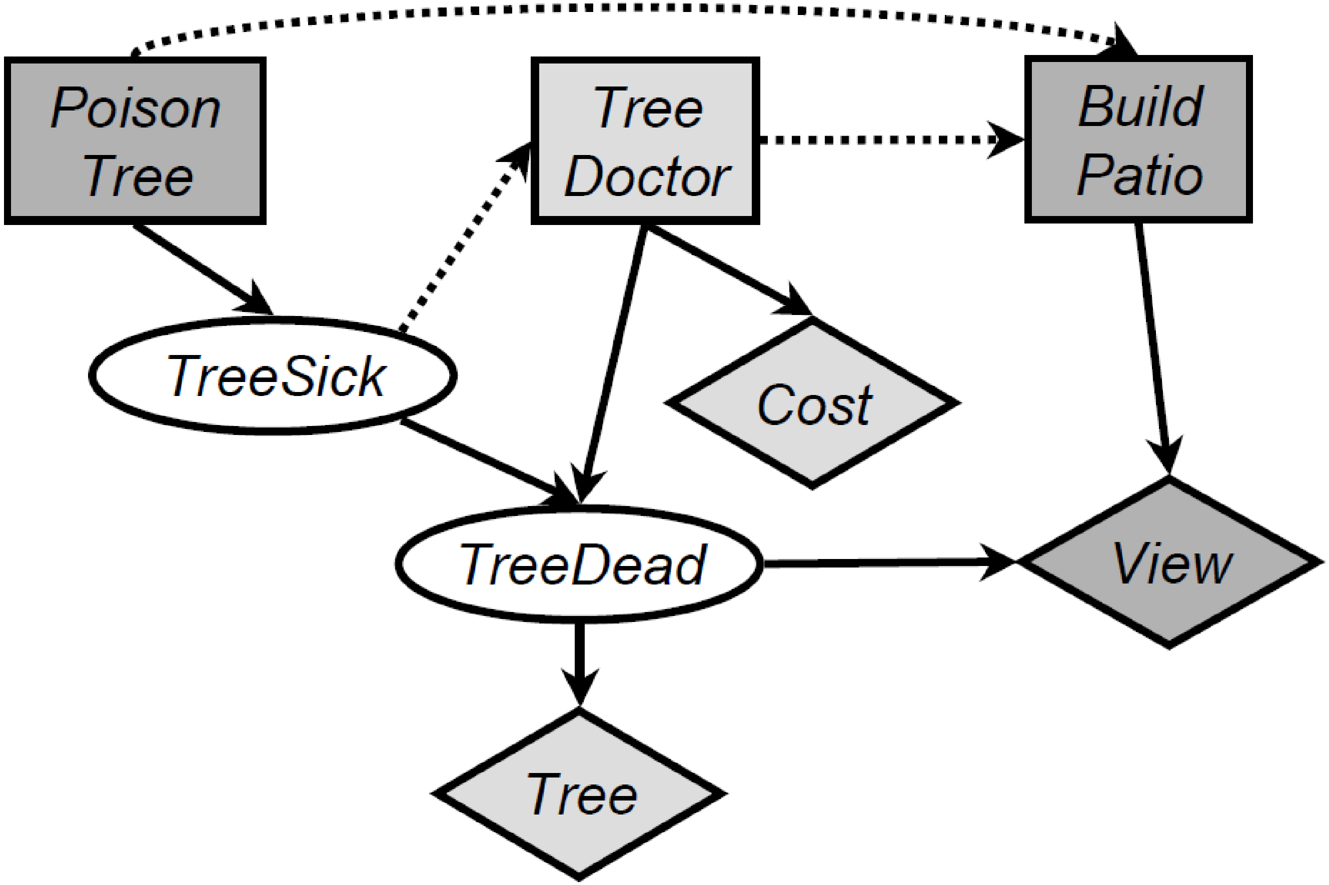
# Multi-agent Inf. Diagrams (Milch and Koller ‘00-01)

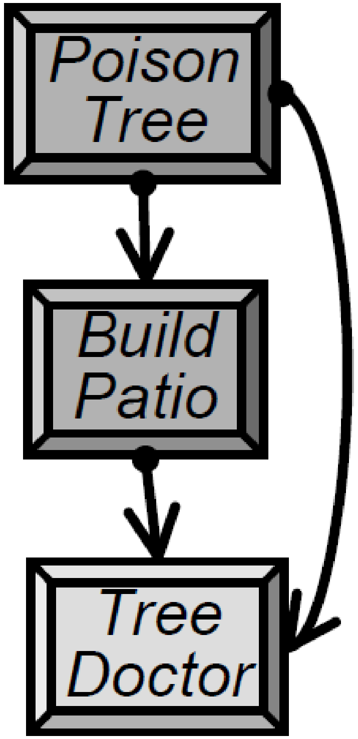
# Motivation

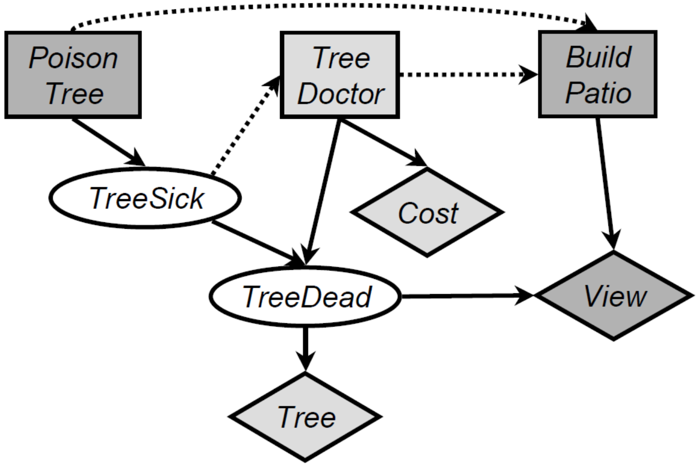
- Settings with explicit self-interest
- Game theory!
- Succinct representation
- Detect structure; allow efficient computation

# TreeKiller example

Example: two agents, Alice (*Poison*, *Build*) and Bob (*Doctor*).







**Relevance graph**

If D relies on D’, there is an edge i
the graph from D to D’.

To optimize for D, need to know decision rule for all children…

solve TreeDoctor;

then BuildPatio;

then PoisonTree

… backward induction (if acyclic relevance graph)

Solve “components” if cycles.

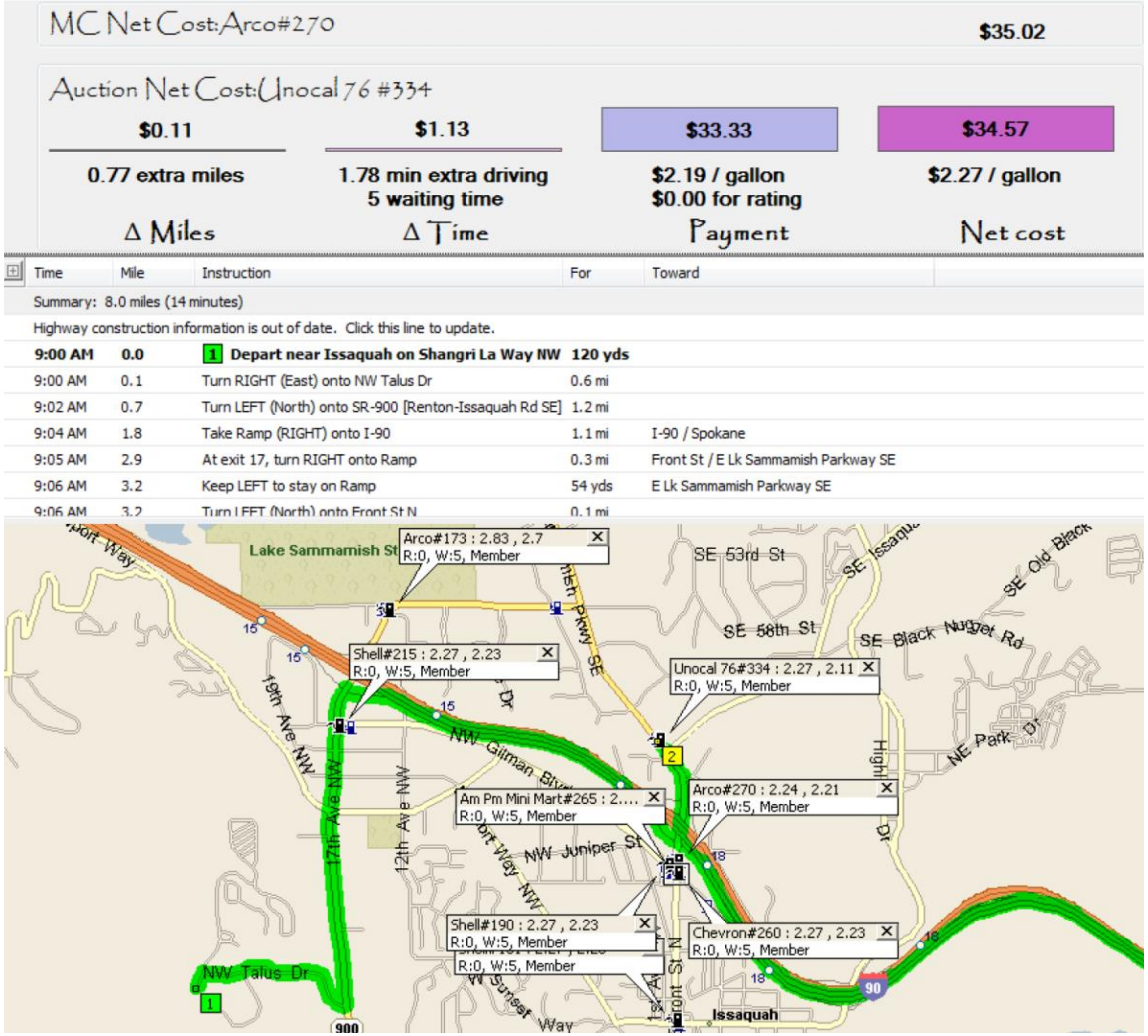
# Modern Examples

- Multi-robot “pick-pack-ship” systems
- Port security (LAX, Boston Harbor, …)
- Smart Power Grid (agents in the home)
- Internet advertising markets (bidding for ads)
- Opportunistic commerce (e.g., agents advising whether to route to get gas…)

# Example: Opportunistic Commerce

(Kamar et al.’08)

- Dynamic matching with location-specific services.



MC Net Cost:Arco#270 $35.02

Auction Net Cost:Unocal 76 #334

| $0.11 | $1.13 | $33.33 | $34.57 |
|---|---|---|---|
| 0.77 extra miles | 1.78 min extra driving<br>5 waiting time | $2.19 / gallon<br>$0.00 for rating | $2.27 / gallon |
| Δ Miles | Δ Time | Payment | Net cost |

| Time | Mile | Instruction | For | Toward |
|---|---|---|---|---|
| Summary: 8.0 miles (14 minutes) | | | | |
| Highway construction information is out of date. Click this line to update. | | | | |
| **9:00 AM** | **0.0** | **1 Depart near Issaquah on Shangri La Way NW** | **120 yds** | |
| 9:00 AM | 0.1 | Turn RIGHT (East) onto NW Talus Dr | 0.6 mi | |
| 9:02 AM | 0.7 | Turn LEFT (North) onto SR-900 [Renton-Issaquah Rd SE] | 1.2 mi | |
| 9:04 AM | 1.8 | Take Ramp (RIGHT) onto I-90 | 1.1 mi | I-90 / Spokane |
| 9:05 AM | 2.9 | At exit 17, turn RIGHT onto Ramp | 0.3 mi | Front St / E Lk Sammamish Parkway SE |
| 9:06 AM | 3.2 | Keep LEFT to stay on Ramp | 54 yds | E Lk Sammamish Parkway SE |
| 9:06 AM | 3.2 | Turn LEFT (North) onto Front St N | 0.1 mi | |

# Course Goals

- Broad and rigorous introduction to the theory, methods and algorithms of multi-agent systems.
- Main intellectual connections with AI, Econ/CS and microeconomic theory
- Emphasize computational perspectives
- Provide a basis for research

- Research seminar--- we’ll read and discuss papers!

# Class participation

- Submit comments on the assigned reading before each class
  - what is the main contribution of the paper?
  - what was the main insight in getting the result?
  - what is not clear to you?
  - what are the most important assumptions, are they limiting?
  - what extensions does this suggest?

- *Start for this Thursday! (Google form…)*

# Student presentations

- You will present 1-2 papers
- Greg Stoddard and I will meet with you to discuss before class
- We will have a joint discussion, driven through your presentation

# Homeworks

- Will be two or three problem sets
- Relatively short (more theoretical than computational)
- Start in around two weeks

# Final Paper

- Study research problem related to class
- Computational, theoretical, experimental or empirical
- Two (3?) people per group (by permission)
- Can be an exposition paper on two related technical papers

- Logistics
  - Submit a proposal 11/12
  - Short presentations 12/3-5
  - Paper due: 12/9

# Grade breakdown

- 20% problem sets
  - two to three of these
- 40% participation
  - Comments, discussion, presentation, Piazza post on something topical
- 40% final project

# Requirements

- CS 181 or CS 182 (or by permission)
- Some background in algorithms, complexity theory, and probability theory
- Background in economic theory useful but not required
- Reasonable level of mathematical sophistication

# Office hours

David Parkes (parkes@eecs.harvard.edu):

- 11.30-12.30p  on 9/3, 9/5 and 9/10 in MD 229
- **Today!!**

- Regularly: 2.30-4pm on Tue/Thur
  - primarily to discuss this week's papers with student presenters

Greg Stoddard (gstoddard@seas.harvard.edu)

- 1.30-2.30p MD 219

# Related AI and Econ/CS Classes

- CS 182 (AI; Fall), CS 181 (ML; Spring)
- CS 186 (EconCS; Spring)

- CS 284r (Networks +AGT; Fall)
- CS 281 (Adv. ML; Fall)
- CS 279 (HCI; Fall)
- CS 280r (Planning; Spring)
- CS 286r (AGT Spring’14, AMD Fall’14)
- CS 289 (Bio-inspired; Spring)

# Next Class

- “Distributed constraint handling and optimization”
- Required Reading before class!

- *Chapter 12* of “MULTIAGENT SYSTEMS;” ed. Gerhard Weiss, MIT Press, 2013, 2nd edition

- Comments on reading due by midnight Wed 9/4
  - ***One paragraph would be fine***
  - *Come prepared to discuss*