Vrije Universiteit Brussel

CoMo
Computational Modeling Lab

# DECENTRALIZED COORDINATION IN MULTI-AGENT SYSTEMS

Mihail “Mike” Mihaylov

# Aim?

to give the intuition just how complex decentralized coordination is

to give the tools necessary to address decentralized coordination problems

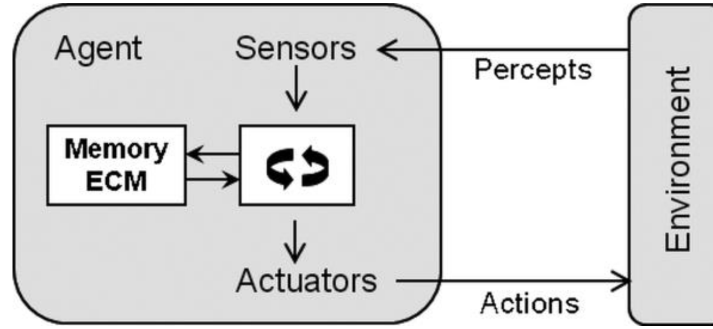
very complex!

to give examples of real-world decentralized coordination problems

Frameworks:

- Multi-agent systems
- Reinforcement Learning
- Collective Intelligence
- Mechanism Design

# Agent

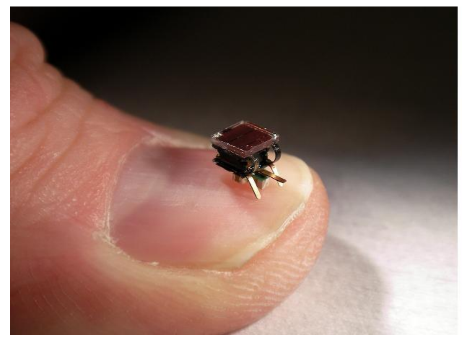



## Autonomous

– has control over own actions
– able to act without human intervention

## Pro-active

– takes initiative
– is opportunistic



## Responsive

– perceives its environment
– responds to changes

## Social

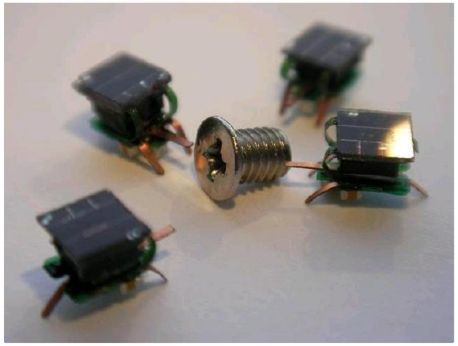
– interacts when appropriate
– helps others

*"Objects do it for free.*
*Agents do it for money."*

# Multi-Agent Systems Framework

**Agents have:**

- incomplete information
- restricted capabilities



**System properties:**

- decentralized control
- **asynchronous** computation

**Communication is:**

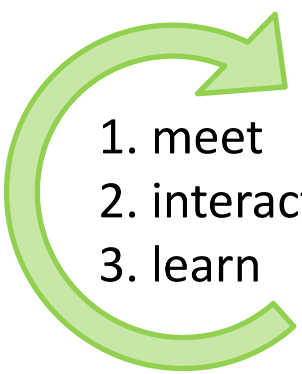
- costly
- delayed
- unreliable

[Wooldridge and Jennings, 1995]

# Context

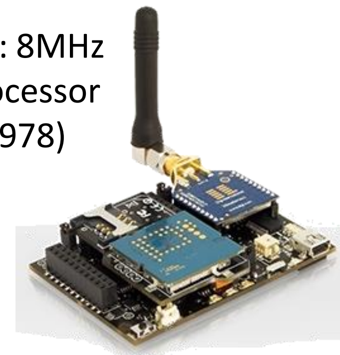
## Constraints:

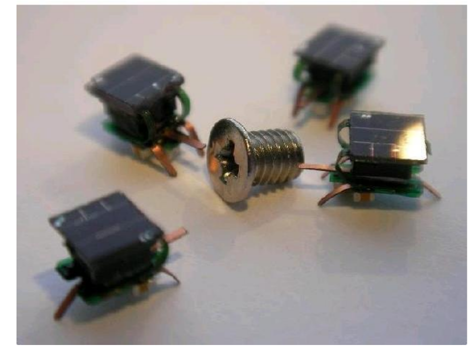
- limited battery
- communication is expensive
- low processing capabilities
- limited knowledge
- decentralized control

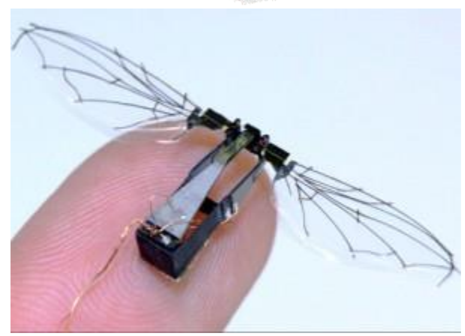


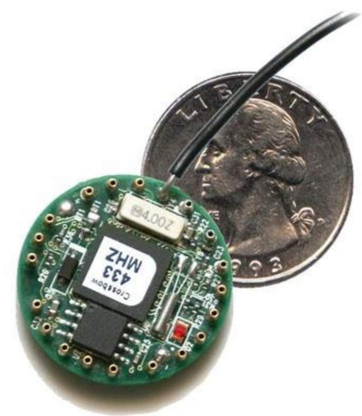

e.g. Reinforcement learning

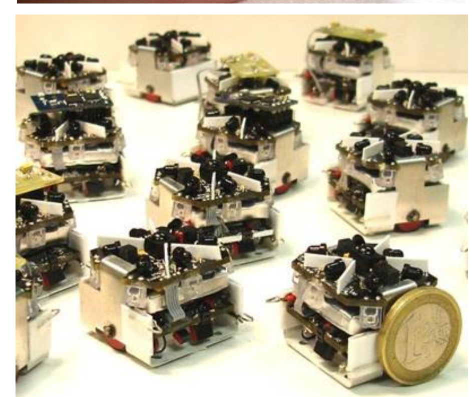
Microcontr.: 8MHz
(as Intel processor
from year 1978)









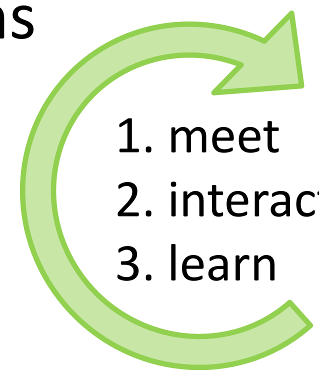


## Decentralized

- central control is unavailable or costly to set up
  (*e.g. WSNs, Swarm robotics*)
- reduce complexity of centralized problems
  (*e.g. Scheduling, Planning*)
- address privacy, self-interest
  (*e.g. Smart grids, Transportation logistics*)

## Coordination

- highly constrained agents
- with limited knowledge
- must work together to solve problems
- learn from repeated interactions



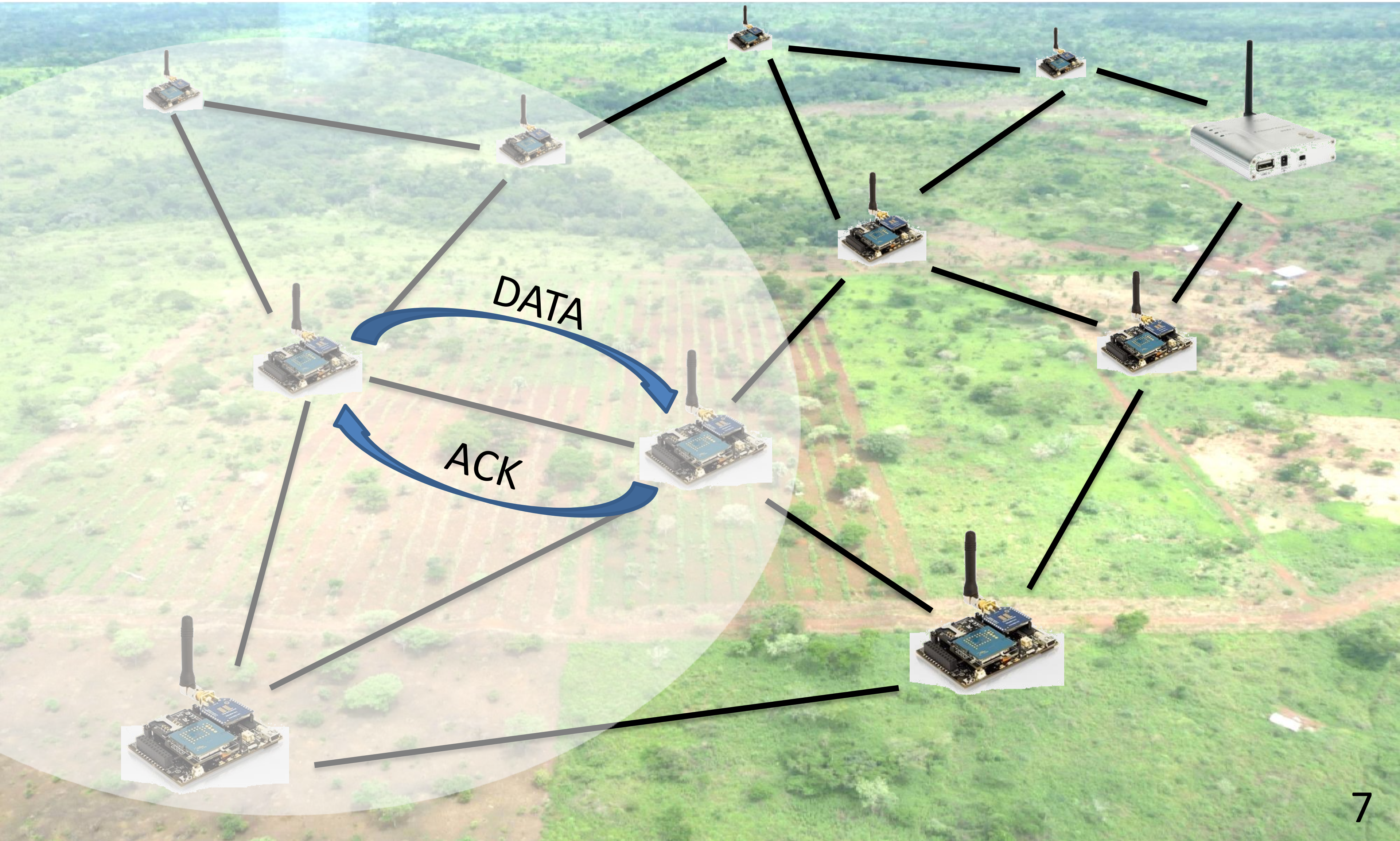

## Agents

- inexpensive
- multi-purpose

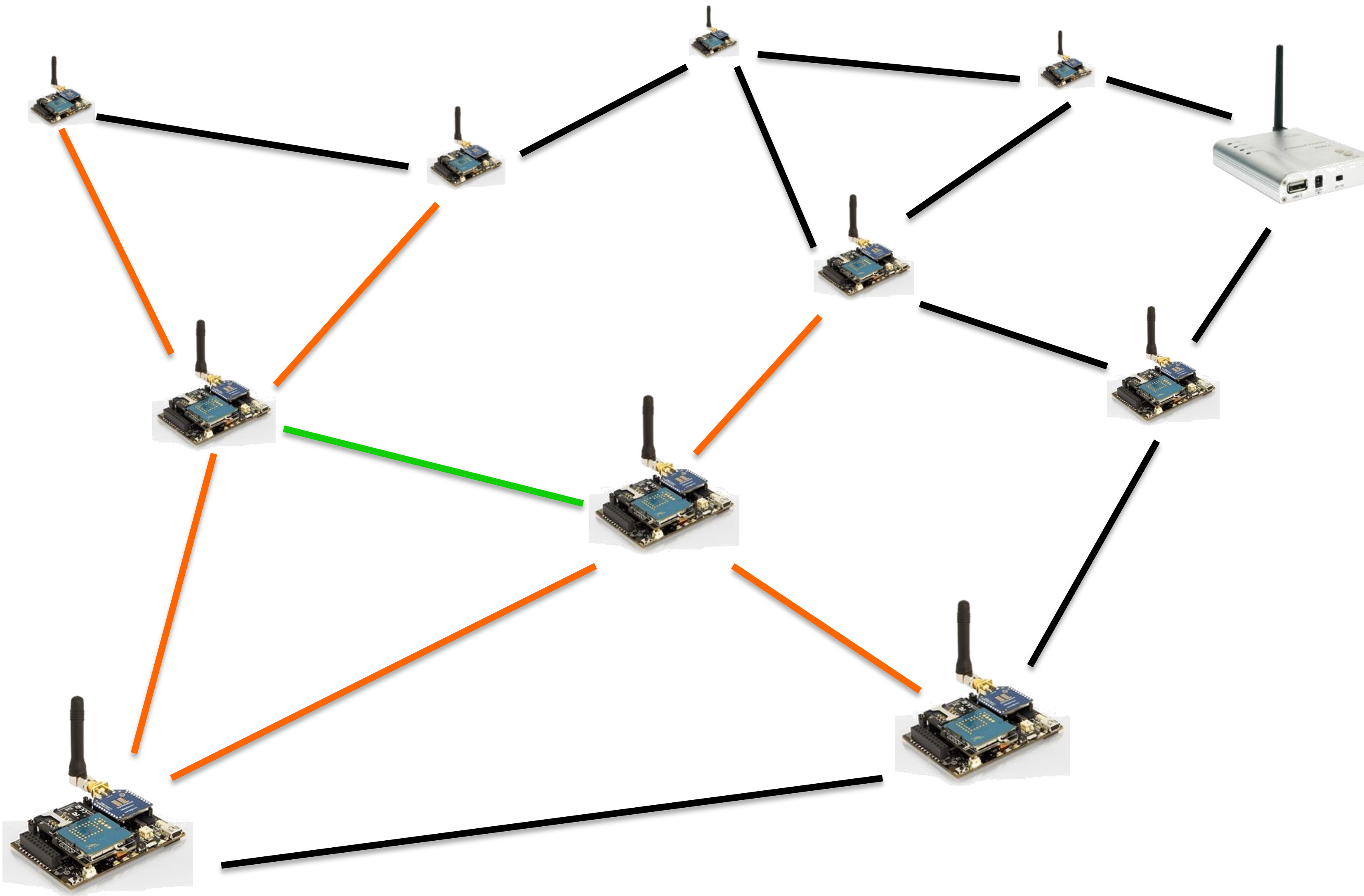
## System

- scalable
- adaptive

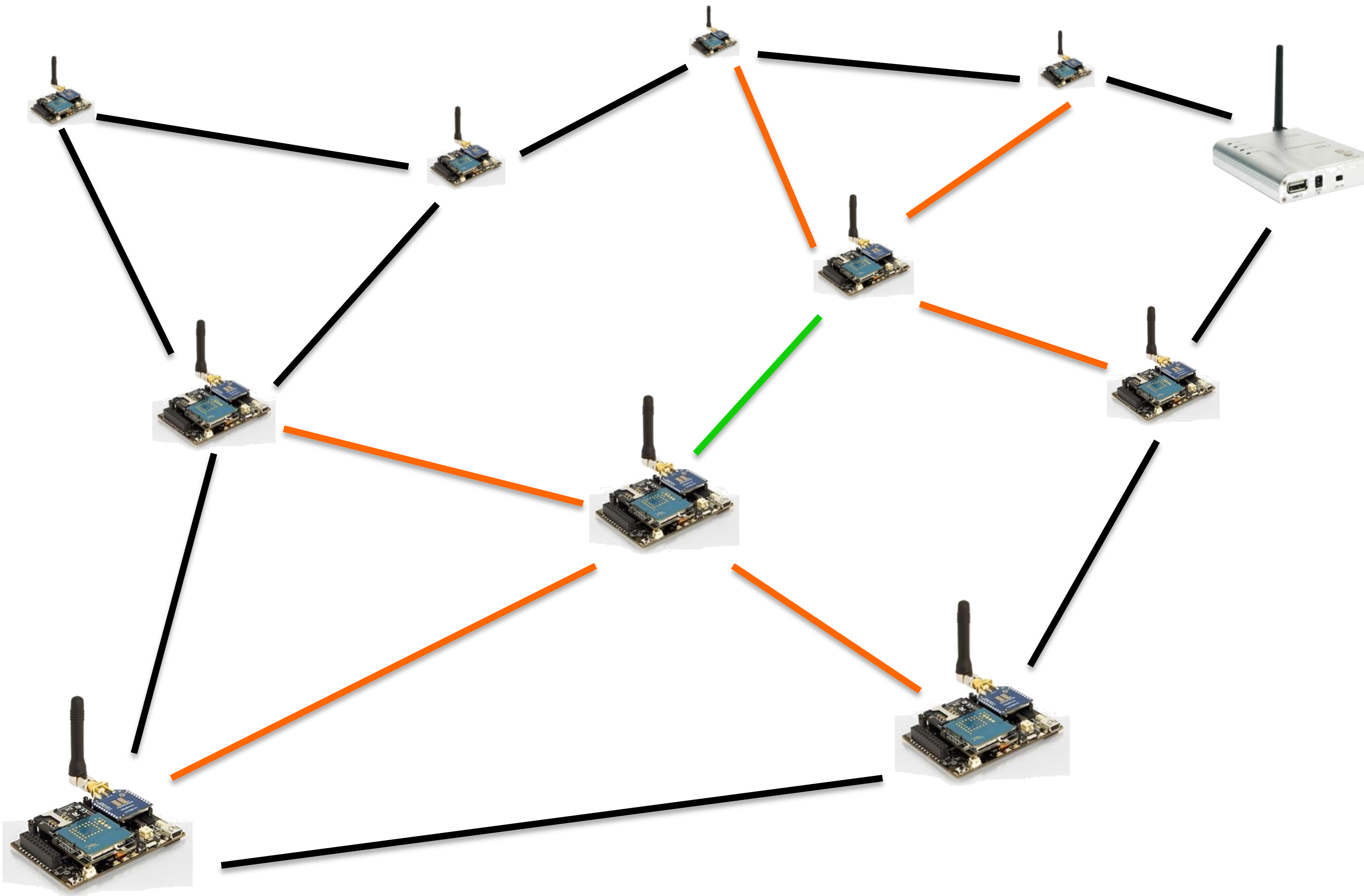
# Decentralized Coordination in Multi-Agent Systems



# Coordination and Anti-Coordination



# Coordination and Anti-Coordination in time



## Problem:

enable decentralized coordination

- highly constrained agents → no complex algorithms
- local interactions → no centralized control
- limited knowledge → no global awareness
- autonomous → no human guidance

## Objective:

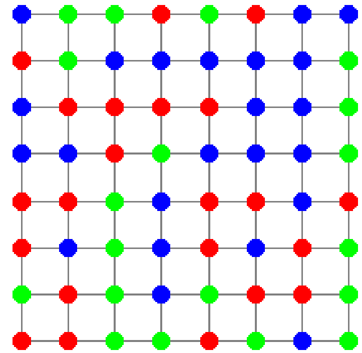
implement a coordination mechanism

- adaptive
- decentralized
- minimal requirements
- minimal overhead

→ perform well in wide range of settings

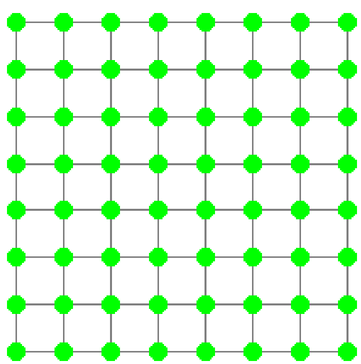
→ few parameters, little memory usage

# Decentralized coordination



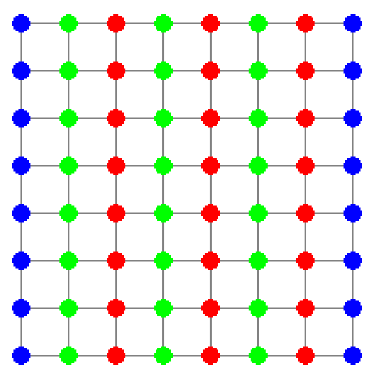
Pure coordination

*e.g. convention emergence*



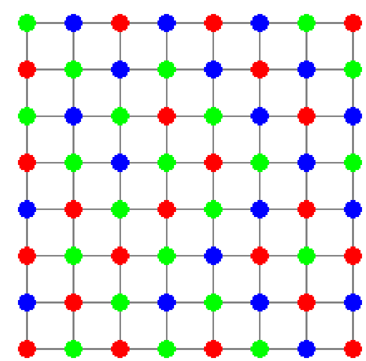
(Anti-)coordination

*e.g. traffic lights*

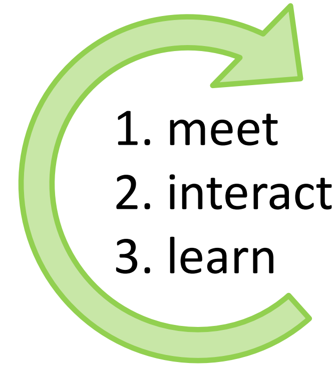


Pure anti-coordination

*e.g. dispersion games*



# Pure coordination

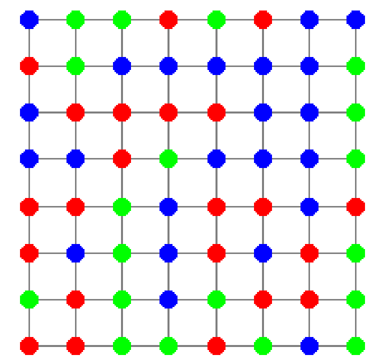



Problem: How to coordinate?

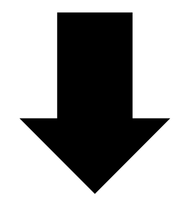
– b/n highly constrained agents

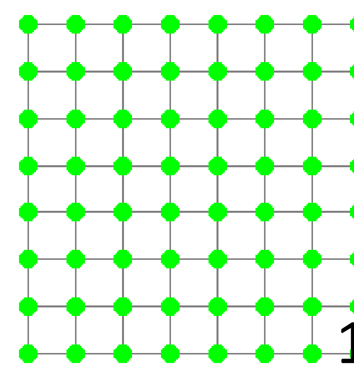
– via local interactions

– with limited knowledge

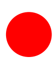


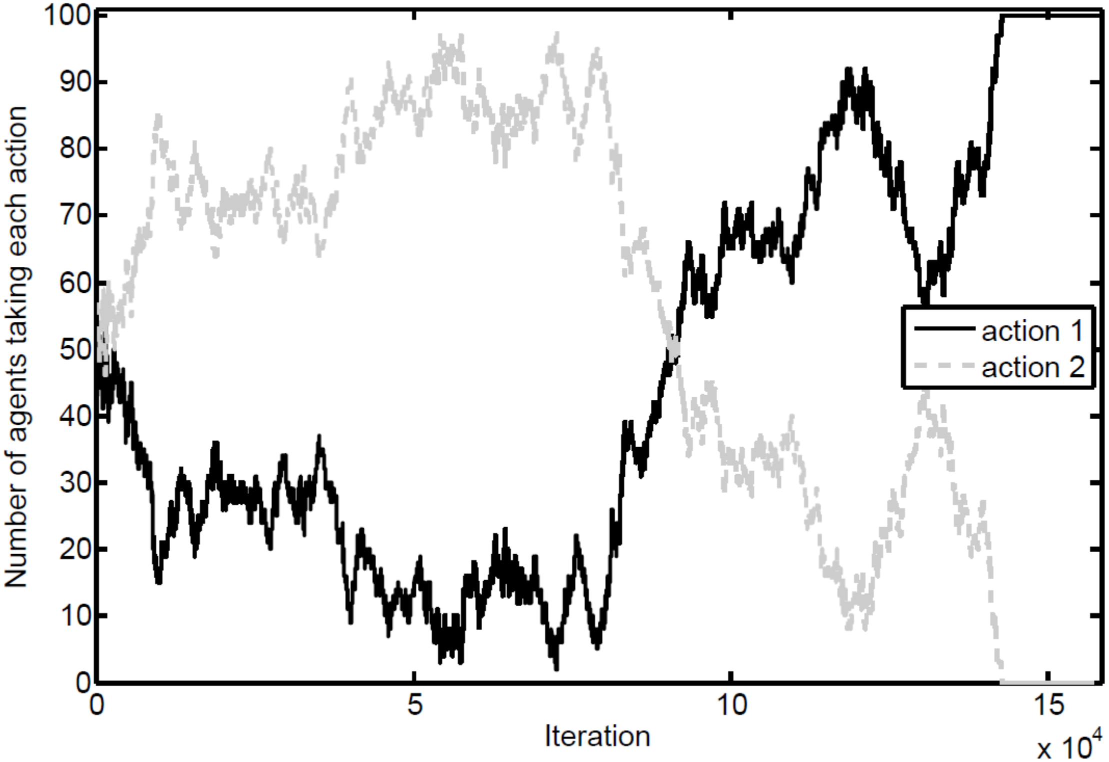
e.g. select the same joint task in robot swarms

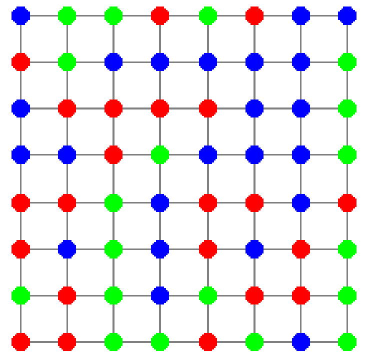




Agent 1

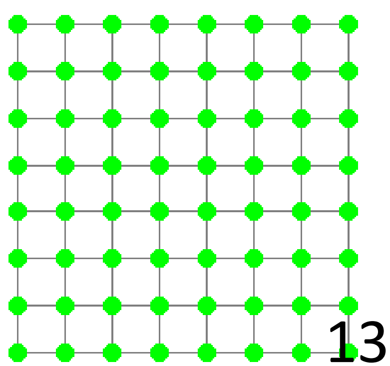
| Agent 2 |  |  |  |
|---|---|---|---|
|  | 1 | 0 | 0 |
|  | 0 | 1 | 0 |
|  | 0 | 0 | 1 |



# Approach: Win-Stay Lose-probabilistic-Shift

e.g. select the same joint task in robot swarms

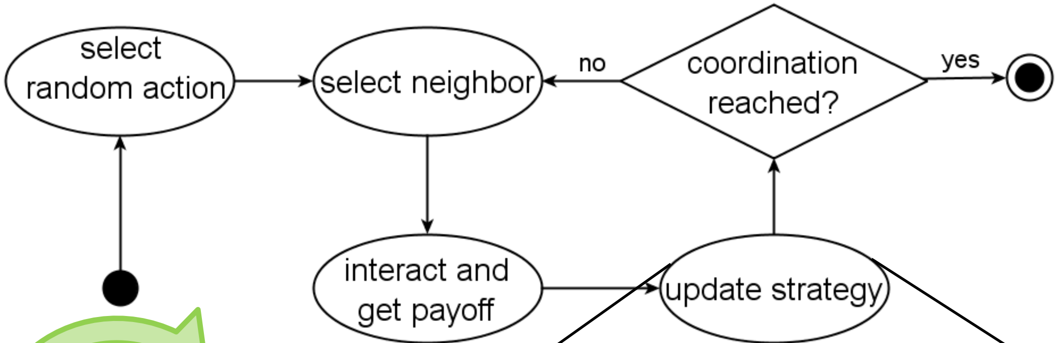


## Requirements:

- decentralized mechanism
- minimal requirements & overhead
- guaranteed full convergence
- absorbing state

# Coordination Game:

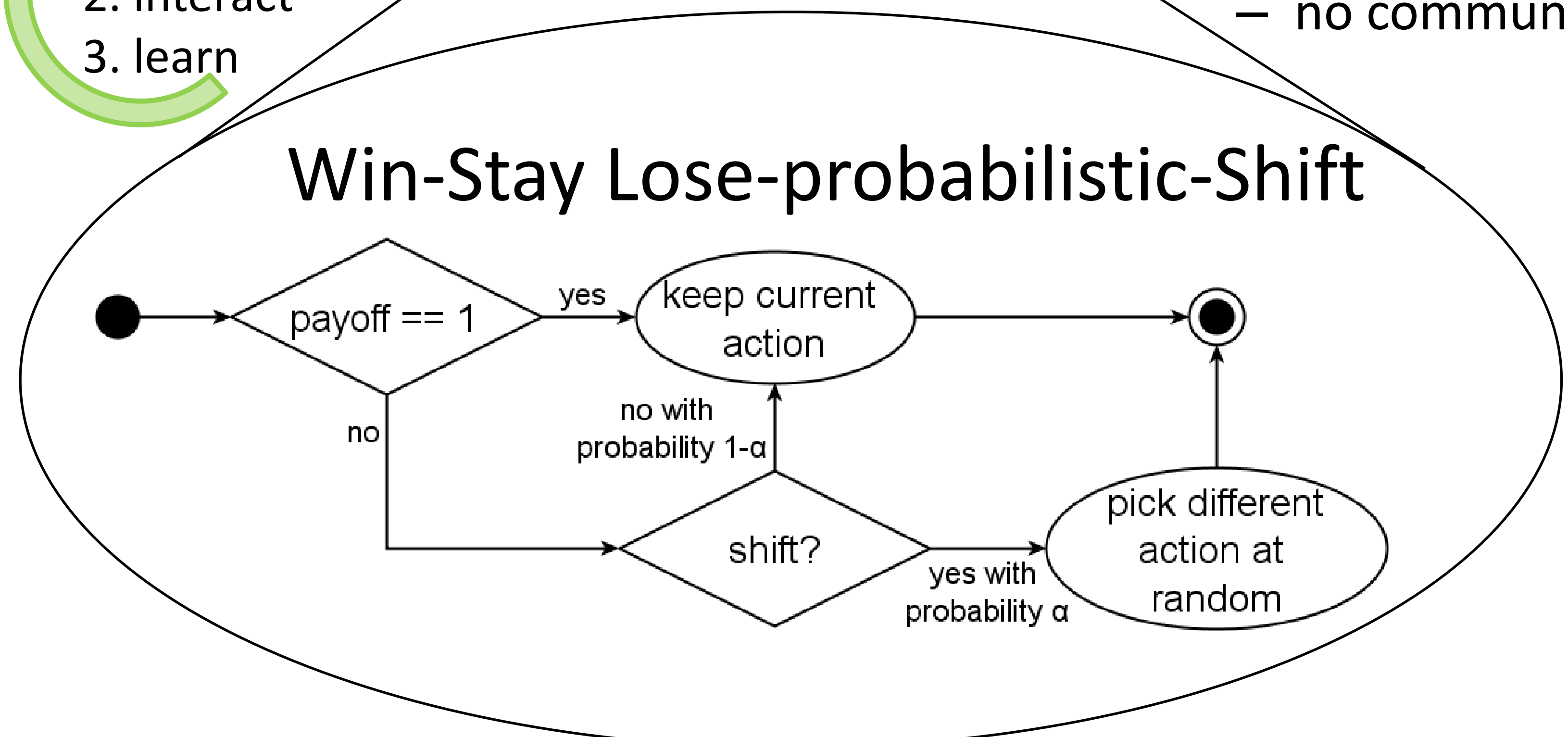
pairwise interactions




1. meet
2. interact
3. learn

$$\text{payoff} = \begin{cases} 1, & \text{if same action} \\ 0, & \text{otherwise} \end{cases}$$

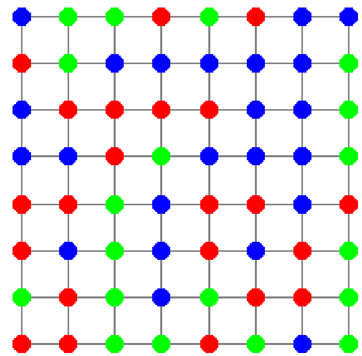
– decentralized approach
– minimal requirements
– no communication



# Conclusions (pure coordination)

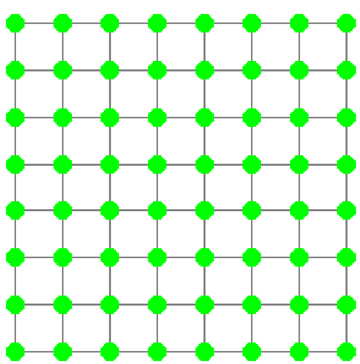
- Pure coordination → difficult, but always possible
- Convergence time → exponential in number of agents and actions
- Denser networks → faster convergence

# Decentralized coordination



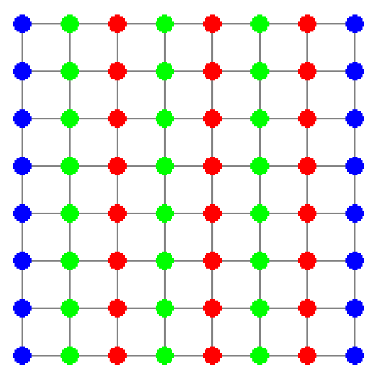
Pure coordination

*e.g. convention emergence*



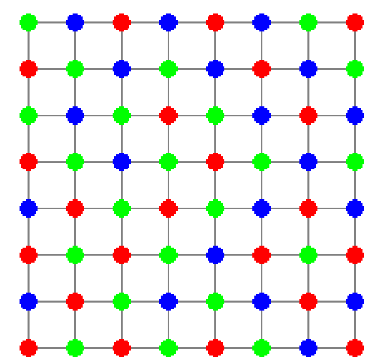
(Anti-)coordination

*e.g. traffic lights*



Pure anti-coordination

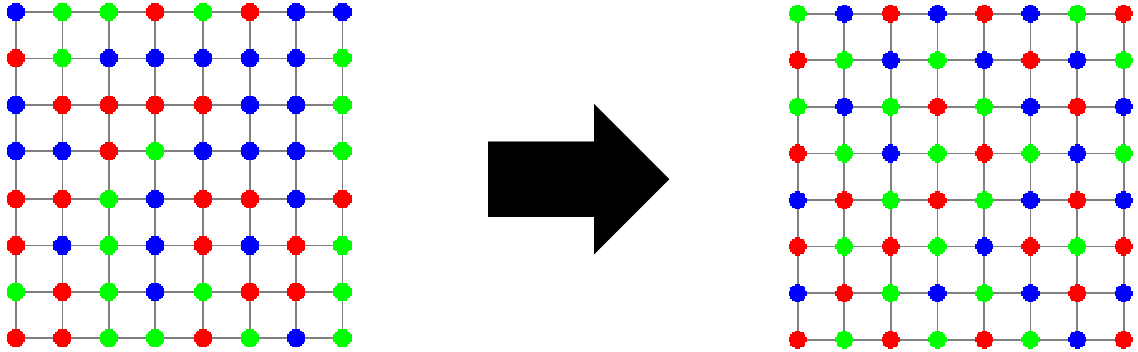
*e.g. dispersion games*



# Anti-coordination

<u>Problem</u>: How to anti-coordinate?

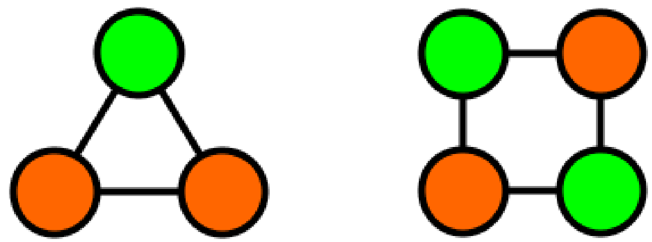
(select an action different than others')



e.g. select different
channels for parallel
communication in WSNs

# Conclusions (anti-coordination)

- Pure anti-coordination → easy but not always feasible

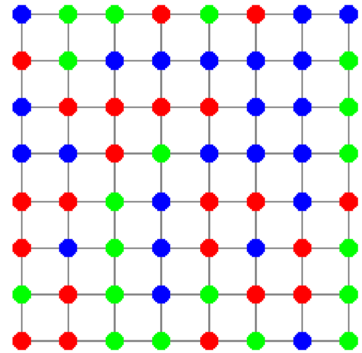


- Convergence time → faster with more actions

- WSLpS: applicable in wide range of scenarios

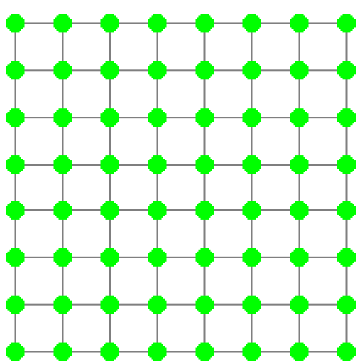
| algorithm | topology: | ring | | | | grid | | | | full | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | actions: | 2 | 3 | 4 | 5 | 2 | 3 | 4 | 5 | 20 | 30 | 40 | 50 |
| WSLpS | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| QL | Grenager *et al*. ‘02 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | |
| Freeze | Grenager *et al*. ‘02 | | | | | | | | | ✓ | ✓ | ✓ | ✓ |
| GaT | Namatame ‘06 | ✓ | | | | ✓ | | | | | | | |

# Decentralized coordination



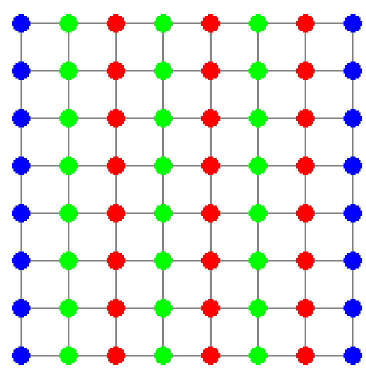
## Pure coordination

*e.g. convention emergence*



## (Anti-)coordination

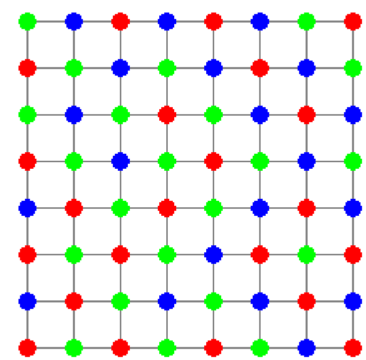
*e.g. traffic lights*



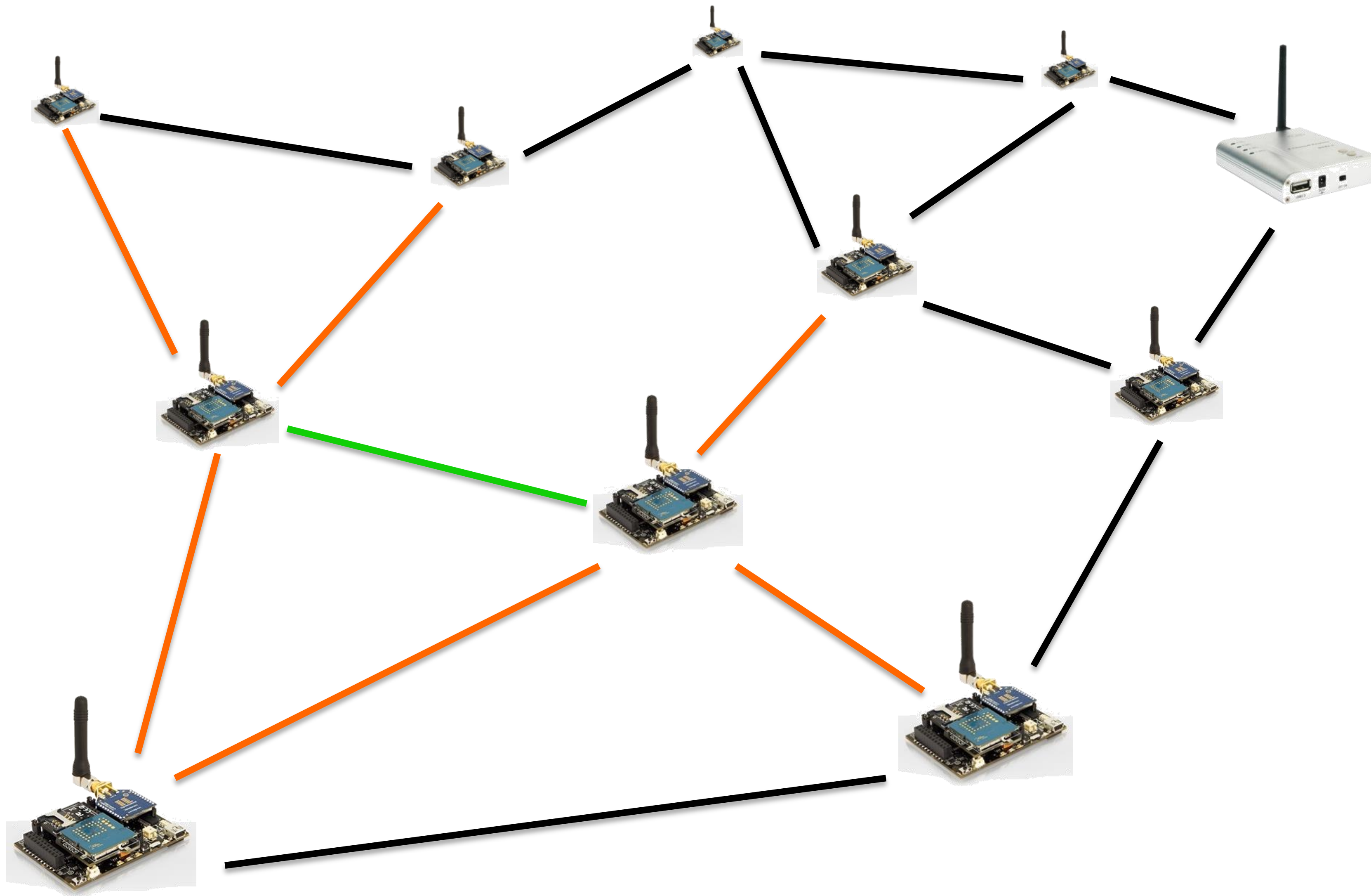
e.g. traffic lights in a city or routing branches in WSNs

## Pure anti-coordination

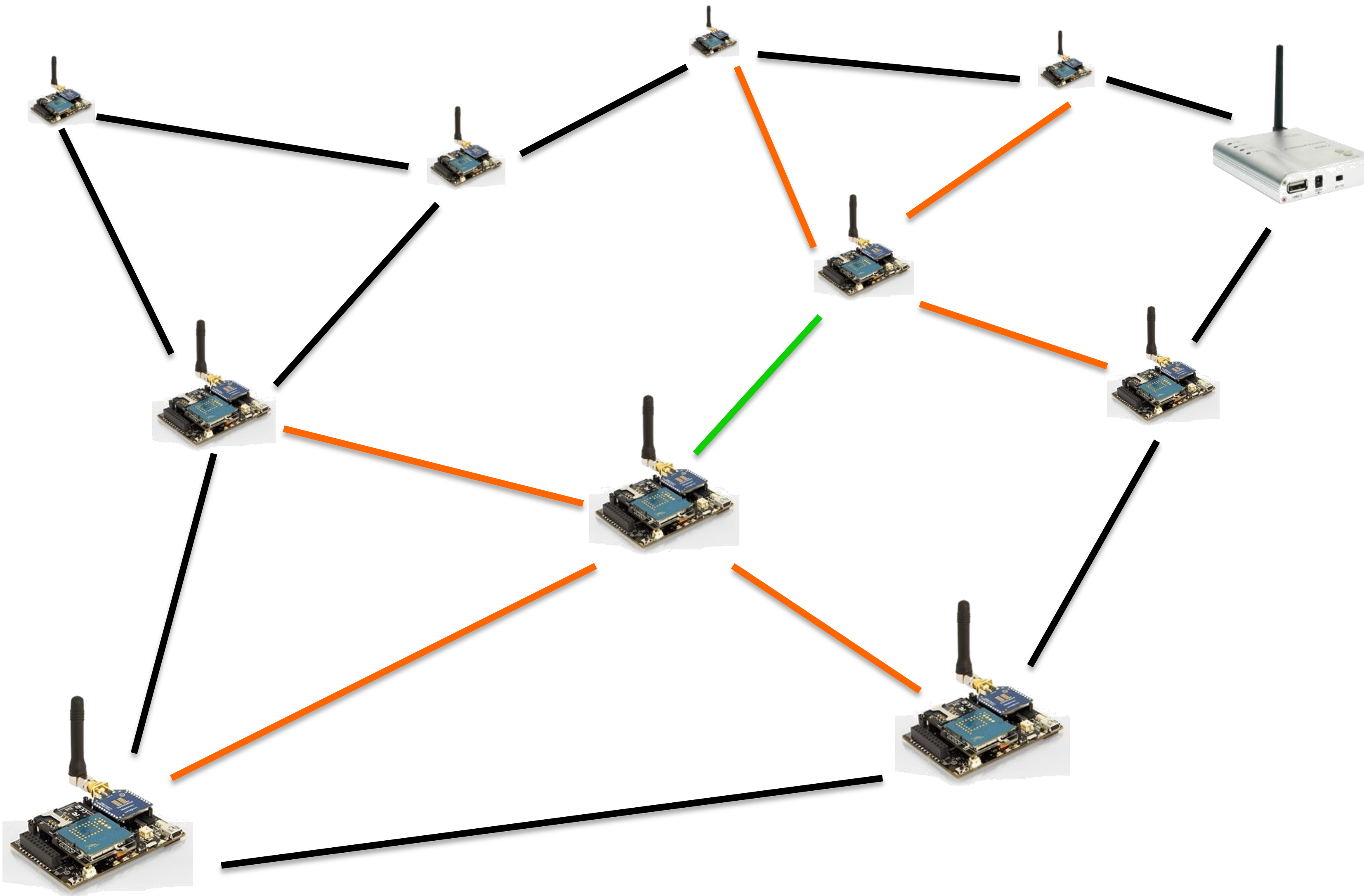
*e.g. dispersion games*



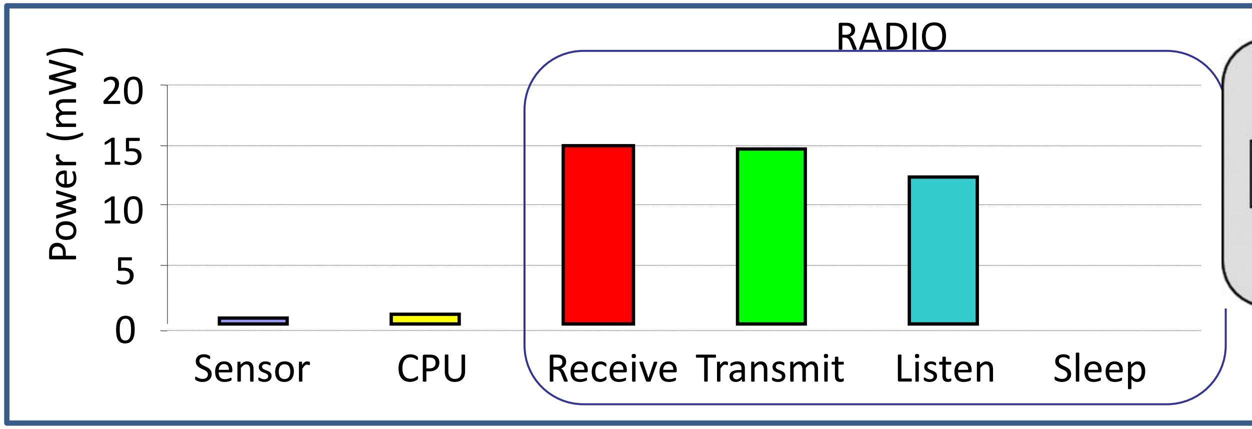
# Coordination and Anti-Coordination

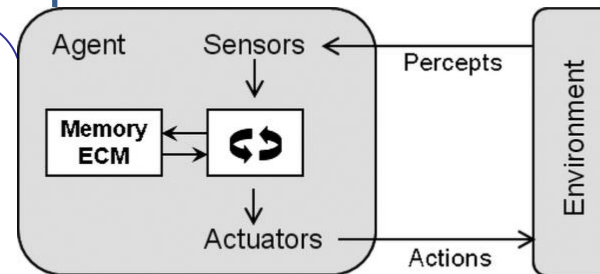


# Coordination and Anti-Coordination in time



# (Anti-)coordination in time: WSNs



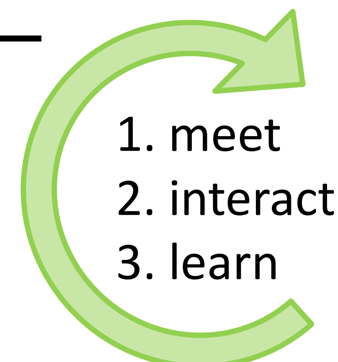

awake
- **Transmit**
  - + Forward messages
  - – Interfere with neighbors
- **Listen**
  - + Receive messages
  - – Waste energy

sleep
- **Sleep**
  - + Save energy
  - – Increase latency

# (Anti-)coordination in time: WSNs

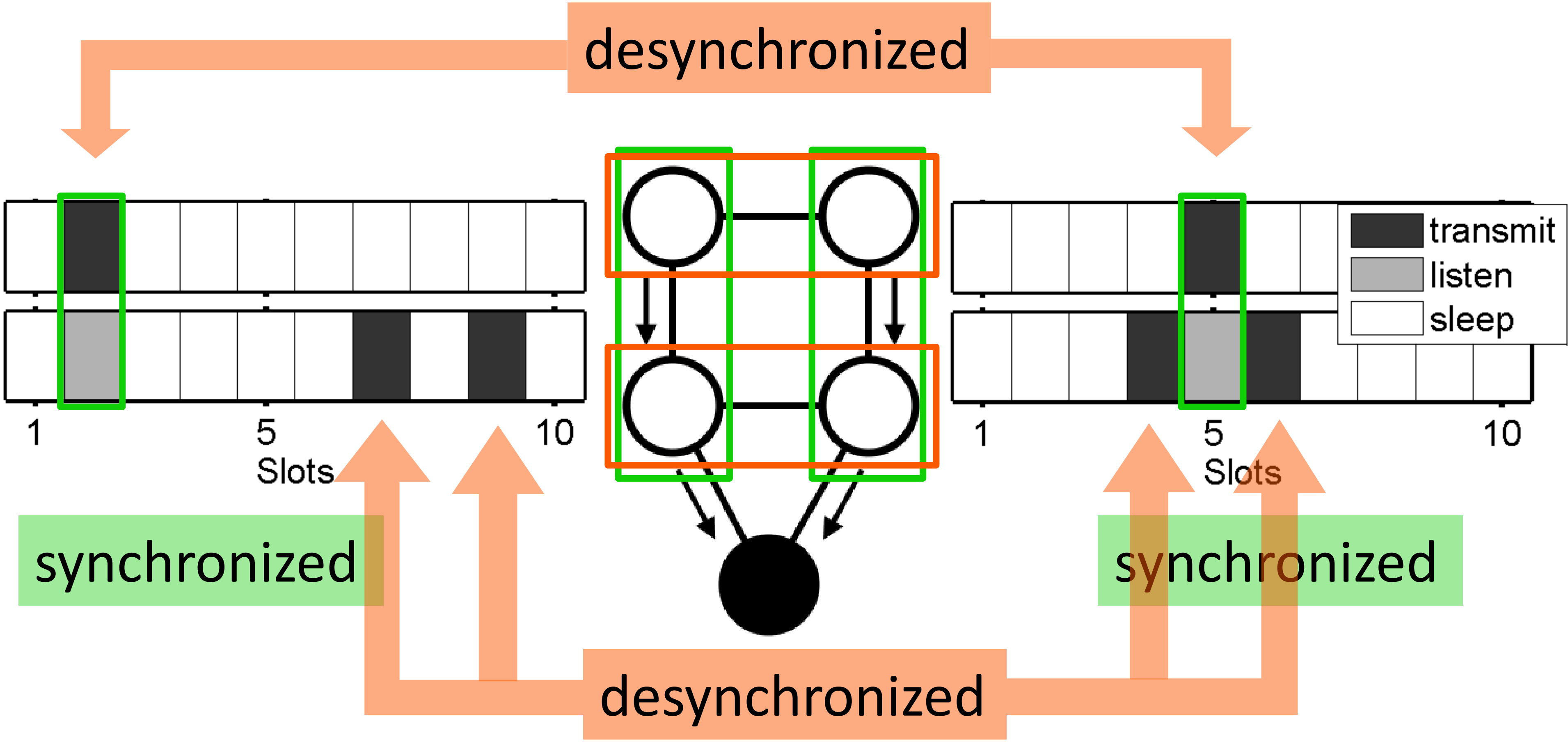
Agents learn by only observing outcome of own actions!

| action | outcome | payoff |
|---|---|---|
| Transmit | ACK received | 1 |
| | no ACK received | 0 |
| Listen | DATA received | 1 |
| | communication overheard | 0 |
| | nothing received | 0 |
| | messages collided | 0 |
| Sleep | saved energy | ? |



# DESYDE

– decentralized approach
– minimal requirements
– no communication overhead



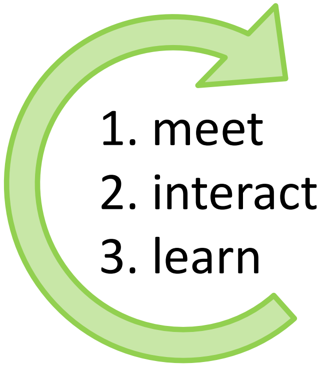
# Conclusions (WSNs)

- Coordination emerges rather than is agreed upon
- Clever coordination mechanism is required to save energy

# Coordination in Time?

## Synchronization
in time

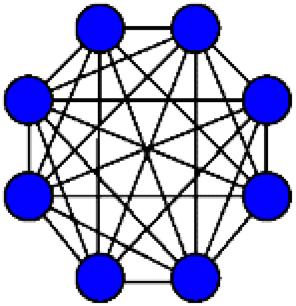
Firefly
(*Photuris lucicrescens*)



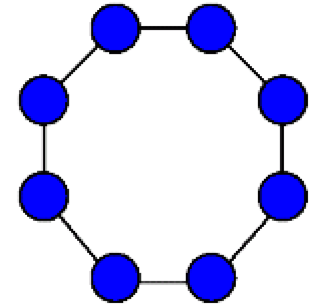

## Desynchronization
in time

Japanese tree frog
(*Hyla japonica*)

Here agents are able to <u>observe</u> actions of <u>all others</u>!



# Graphical games (Network games)



**Normal form**

- any agent interacts with any other agent
- payoffs depend on actions of all agents
- representation: exponential in the number of players

**Graphical form**

- only neighbors in the graph can interact
- payoffs depend on actions of neighbors
- representation: exponential in size of largest neighborhood

# Outlook

| Covered topics | Other topics | Framework |
|---|---|---|
| Common interest game | Conflicting interest game | Cooperative game theory |
| Fully cooperative agents | Self-interested agents | Mechanism design |
| Static topology | Dynamic topology | Evolutionary game theory |
| Wireless sensor networks | Collaborative platforms | Collective intelligence |

# Collective Intelligence Framework

**Components**:

- private utility → measures performance of individual agents
- world utility → measures performance of the entire system

**Objective**:

- align private utility with world utility

e.g.: optimizing the routing of internet traffic

**Challenges**:

- compute world utility in a decentralized manner

[Wolpert and Tumer, 2003]

# Mechanism Design Framework

<u>Agents</u>:

- are self-interested
- have private information
- maximize utility functions

<u>Objective</u>:

- implement a protocol that achieves designer's goals, despite agents' self interest

Game theory:

– *"Given a game, what is a rational strategy for an agent?"*

Mechanism design:

– *"Given that agents are rational, how should we design the game?"*

e.g.: designing an electronic auction

[Conitzer and Sandholm, 2003]

# Conclusions

Simple learning techniques work surprisingly well.

- \+ less parameters to tune
- \+ quite generic, wider application
- – lower representational power

Decentralized solutions are a powerful paradigm.

- \+ lower computational complexity & communication overhead
- \+ no single-point-of-failure problems
- – difficult to organize

Multi-agent systems is a quite useful framework.

- \+ suitable representation of decentralized problems
- \+ scalable, fault-tolerant
- – easy to overdo