A Theory Approach to Local-to-Global Algorithms in Spatial Multi-Agent Systems

CS266, Fall 2007
Dan Yamins

Session 1: 12.04.2007
Spatial Multi-agent Systems

a space
Spatial Multi-agent Systems

a space with agents embedded in the space
Spatial Multi-agent Systems

a space with agents embedded in the space

local information and processing
Spatial Multi-agent Systems

a space with agents embedded in the space

local information and processing
globally defined tasks
biology:

drosophila embryo
biology:

*drosophila* embryo
biology:

drosophila embryo
biology:

drosophila embryo

polistes nest
biology:

*droso*phila embryo

polistes nest

white ibis flock
engineering:
Sensor Networks
engineering:

Sensor Networks

McLurkin iRobot Swarm
engineering:

Sensor Networks

McLurkin iRobot Swarm

Saul Griffith’s self-folding structures
engineering:

Sensor Networks

McLurkin iRobot Swarm

Saul Griffith’s self-folding structures

Butera’s “paintable computer” concept
Challenges ...
Challenges ...

- It's hard to translate Global imperatives into local actions:
Challenges ...

- It’s hard to translate Global imperatives into local actions:
  - G to L: description level mismatch
Challenges ...

• It’s hard to translate Global imperatives into local actions:
  • G to L: description level mismatch
  • L to G: inscrutable complexity
Challenges ...

- It’s hard to translate Global imperatives into local actions:
  - G to L: description level mismatch
  - L to G: inscrutable complexity

- Desire for provable and quantifiable robustness
Challenges ...

- It’s hard to translate Global imperatives into local actions:
  - G to L: description level mismatch
  - L to G: inscrutable complexity

- Desire for provable and quantifiable robustness

- What agent resource capacities are required to solve a given task? Is the global task even locally solvable at all?
Challenges ...

- It’s hard to translate Global imperatives into local actions:
  - G to L: description level mismatch
  - L to G: inscrutable complexity

- Desire for provable and quantifiable robustness

- What agent resource capacities are required to solve a given task? Is the global task even locally solvable at all?

- What does thinking of natural spatial computers \textit{qua} computers tell us scientically?
Challenges ...

- It’s hard to translate Global imperatives into local actions:
  - G to L: description level mismatch
  - L to G: inscrutable complexity

- Desire for provable and quantifiable robustness

- What agent resource capacities are required to solve a given task? Is the global task even locally solvable at all?

- What does thinking of natural spatial computers *qua* computers tell us scientifically?

... the Need for A Theory
Specific Problems ...
Specific Problems ...

- **A Description Problem:** What are appropriate formal models for spatial multi-agent systems? For the agents themselves? For local rules? For global tasks? (Today)
Specific Problems ...

- **A Description Problem:** What are appropriate formal models for spatial multi-agent systems? For the agents themselves? For local rules? For global tasks? (Today)

- **An Existence Problem:** When do given global tasks even have robust local rule solutions in the first place? What features make a global problem robustly solvable? (Session 2)
Specific Problems ...

- **A Description Problem:** What are appropriate formal models for spatial multi-agent systems? For the agents themselves? For local rules? For global tasks? (Today)

- **An Existence Problem:** When do given global tasks even have robust local rule solutions in the first place? What features make a global problem robustly solvable? (Session 2)

- **A Construction Problem:** Given that problem is solvable, can we algorithmically *construct* a generic procedure for producing solutions? (Session 3)
Specific Problems ...

- **A Description Problem:** What are appropriate formal models for spatial multi-agent systems? For the agents themselves? For local rules? For global tasks? (Today)

- **An Existence Problem:** When do given global tasks even have robust local rule solutions in the first place? What features make a global problem robustly solvable? (Session 2)

- **A Construction Problem:** Given that problem is solvable, can we algorithmically construct a generic procedure for producing solutions? (Session 3)

- **A Resources Problem:** The main parameters of a spatial multi-agent system are its *communications capacity* and the *amount of agent internal memory*. Can we trade off between the two? How do the answers to the other problems scale? (Session 4)
Specific Problems ...

- **A Description Problem:** What are appropriate formal models for spatial multi-agent systems? For the agents themselves? For local rules? For global tasks? (Today)

- **An Existence Problem:** When do given global tasks even have robust local rule solutions in the first place? What features make a global problem robustly solvable? (Session 2)

- **A Construction Problem:** Given that problem is solvable, can we algorithmically construct a generic procedure for producing solutions? (Session 3)

- **A Resources Problem:** The main parameters of a spatial multi-agent system are its *communications capacity* and the *amount of agent internal memory*. Can we trade off between the two? How do the answers to the other problems scale? (Session 4)

... Global-to-Local compilation.
Definition. An underlying space \( \mathcal{G} \) of (partially) directed graphs.
The Model: Static Configurations

**Definition.** An underlying space is a set \( G \) of (partially) directed graphs.
The Model: Static Configurations

**Definition.** An underlying space is a set $G$ of (partially) directed graphs.

Undirected Lines
The Model: Static Configurations

**Definition.** An underlying space is a set $G$ of (partially) directed graphs.

**Undirected Lines**

**Directed Lines**
The Model: Static Configurations

**Definition.** An underlying space is a set $\mathcal{G}$ of (partially) directed graphs.

- **Undirected Lines**

- **Directed Lines**

- **Ring Lattices**
Definition. An underlying space is a set $\mathcal{G}$ of (partially) directed graphs.
The Model: Static Configurations

**Definition.** An underlying space is a set $G$ of (partially) directed graphs.

Usually is a *large or infinite* set of configurations.
**Definition.** An underlying space is a set $\mathcal{G}$ of (partially) directed graphs.
The Model: Static Configurations

**Definition.** An underlying space is a set $\mathcal{G}$ of (partially) directed graphs.
Definition. An underlying space is a set $G$ of (partially) directed graphs.
Definition. An underlying space is a set $\mathcal{G}$ of (partially) directed graphs.
The Model: Static Configurations

**Definition.** An underlying space is a set $\mathcal{G}$ of (partially) directed graphs.
Definition. An underlying space is a set $\mathcal{G}$ of (partially) directed graphs.
**Definition.** An underlying space is a set $\mathcal{G}$ of (partially) directed graphs.
**The Model: Static Configurations**

**Definition.** An underlying space is a set \( G \) of (partially) directed graphs.

**Triangulated Sphere**
**Definition.** An underlying space is a set $\mathcal{G}$ of (partially) directed graphs.
The Model: Static Configurations

**Definition.** An underlying space is a set $\mathcal{G}$ of (partially) directed graphs.
The Model: Static Configurations

**Definition.** An underlying space is a set $G$ of (partially) directed graphs.

**Definition.** Configurations are labelings of graphs in $G$ with elements of some “state set” $S$. 
The Model: Static Configurations

**Definition.** An **underlying** space is a set $G$ of (partially) directed graphs.

**Definition.** Configurations are labelings of graphs in $G$ with elements of some “state set” $S$.

State labels represent the agent’s internal states.
The Model: Static Configurations

Definition. An underlying space is a set $G$ of (partially) directed graphs.

Definition. Configurations are labelings of graphs in $G$ with elements of some “state set” $S$.

State labels represent the agent’s internal states.
The Model: Static Configurations

**Definition.** An underlying space is a set $\mathcal{G}$ of (partially) directed graphs.

**Definition.** Configurations are labelings of graphs in $\mathcal{G}$ with elements of some “state set” $S$.

State labels represent the agent’s internal states.

**Definition.** $B_r(a, X)$ is the local ball of radius $r$ around agent $a$ in configuration $X$. 
The Model: Static Configurations

**Definition.** An underlying space is a set $\mathcal{G}$ of (partially) directed graphs.

**Definition.** Configurations are labelings of graphs in $\mathcal{G}$ with elements of some “state set” $S$.

State labels represent the agent’s internal states.

**Definition.** $B_r(a, X)$ is the local ball of radius $r$ around agent $a$ in configuration $X$. 
**The Model: Static Configurations**

**Definition.** An underlying space is a set $\mathcal{G}$ of (partially) directed graphs.

**Definition.** Configurations are labelings of graphs in $\mathcal{G}$ with elements of some “state set” $S$.

State labels represent the agent’s internal states.

**Definition.** $B_r(a, X)$ is the local ball of radius $r$ around agent $a$ in configuration $X$.

$B_2(6, X)$ for $X = 1$-D configuration shown above
The Model: Local Rule Dynamics
Definition. A radius $R$ local rule is a look-up table that maps $R$-ball configurations to “updated state” choices. Formally,

$$F : B_{r,S} \rightarrow S.$$
The Model: Local Rule Dynamics

**Definition.** A radius $R$ local rule is a look-up table that maps $R$-ball configurations to "updated state" choices. Formally,

$$F : B_{r,S} \longrightarrow S.$$
The Model: Local Rule Dynamics

**Definition.** A radius $R$ local rule is a look-up table that maps $R$-ball configurations to “updated state” choices. Formally,

$$F : \mathcal{B}_{r,S} \rightarrow S.$$ 

Thought of as “agent-based local programs”.
Sequences of agent calls can be synchronous or asynchronous.
Sequences of agent calls can be synchronous or asynchronous.

- Completely Synchronous: all agents called at all timesteps.
Sequences of agent calls can be synchronous or asynchronous.

- Completely Synchronous: all agents called at all timesteps.
- Completely asynchronous: only one agent called per timestep, and no other restriction, except liveness.
Sequences of agent calls can be synchronous or asynchronous.

- Completely Synchronous: all agents called at all timesteps.
- Completely asynchronous: only one agent called per timestep, and no other restriction, except liveness.

(liveness: no agent ever stops being called forever)
Sequences of agent calls can be synchronous or asynchronous.

- Completely Synchronous: all agents called at all timesteps.

- Completely asynchronous: only one agent called per timestep, and no other restriction, except liveness.

  (liveness: no agent ever stops being called forever)

- k-bounded asynchronous: only one agent called per timestep, and no agent called k+1 times before all others called once.
Sequences of agent calls can be synchronous or asynchronous.

- Completely Synchronous: all agents called at all timesteps.
- Completely asynchronous: only one agent called per timestep, and no other restriction, except liveness.
  
  \[(\text{liveness: no agent ever stops being called forever})\]

- k-bounded asynchronous: only one agent called per timestep, and no agent called \(k+1\) times before all others called once.

**Definition.** A size-\(n\) call sequence is a call sequence acting on an \(n\)-agent configuration. A timing model is a set of call sequences for each size \(n\).
The Model: Local Rule Dynamics
The Model: Local Rule Dynamics

**Definition.** The trajectory \( \{F^n_s(X_0)\} \) is generated by iterating \( F \) from the initial condition \( X_0 \), calling agents as specified by call sequence \( s \).
**Definition.** The trajectory \( \{F^n_s(X_0)\} \) is generated by iterating \( F \) from the initial condition \( X_0 \), calling agents as specified by call sequence \( s \).
The Model: Patterns
The Model: Patterns

Definition. A pattern is a set of configurations.
Definition. A pattern is a set of configurations.
The Model: Patterns

**Definition.** A pattern is a set of configurations.
The Model: Patterns

**Definition.** A pattern is a set of configurations.

Repeat Patterns
The Model: Patterns

**Definition.** A pattern is a set of configurations.

Repeat Patterns
Definition. A pattern is a set of configurations.
The Model: Patterns
The Model: Patterns

**Definition.** *A pattern is a set of configurations.*
The Model: Patterns

**Definition.** A pattern is a set of configurations.

Proportionate Patterns
**Definition.** A pattern is a set of configurations.
The Model: Patterns

**Definition.** A pattern is a set of configurations.

Proportionate Patterns

[Diagram of proportionate patterns with red and green circles connected by arrows]

Thursday, November 28, 13
The Model: Patterns

**Definition.** A pattern is a set of configurations.
The Model: Patterns

**Definition.** A pattern is a set of configurations.
The Model: Patterns

**Definition.** A pattern is a set of configurations.

Proportionate Patterns

Other patterns
Definition. A local rule $F$ is a robust solution to pattern $T$ in timing model $\mathcal{S}$ if,
The Model: Robust Solutions

**Definition.** A *local rule* $F$ is a robust solution to pattern $T$ in timing model $S$ if,
The Model: Robust Solutions

**Definition.** A local rule $F$ is a robust solution to pattern $T$ in timing model $S$ if, for all sizes $n$
The Model: Robust Solutions

**Definition.** A local rule $F$ is a robust solution to pattern $T$ in timing model $S$ if, for all sizes $n$ and all configurations $X$ of size $n$
The Model: Robust Solutions

**Definition.** A local rule $F$ is a robust solution to pattern $T$ in timing model $S$ if, for all sizes $n$ and all configurations $X$ of size $n$ and all call sequences on $X$.
The Model: Robust Solutions

**Definition.** A local rule $F$ is a robust solution to pattern $T$ in timing model $S$ if, for all sizes $n$ and all configurations $X$ of size $n$ and all call sequences on $X$

$$\lim_{n \to \infty} F_s^n(X)$$
The Model: Robust Solutions

**Definition.** A local rule $F$ is a robust solution to pattern $T$ in timing model $S$ if, for all sizes $n$ and all configurations $X$ of size $n$ and all call sequences on $X$

$$\lim_{n \to \infty} F^n_s(X)$$

whenever $T$ contains one more instances of size $n$. 
The Model: Robust Solutions

**Definition.** *A local rule* $F$ *is a robust solution to pattern* $T$ *in timing model* $S$ *if, for all sizes* $n$ *and all configurations* $X$ *of size* $n$ *and all call sequences on* $X$

$$
\lim_{n \to \infty} F_s^n(X)
$$

*whenever* $T$ *contains one more instances of size* $n$. 
The Model: Robust Solutions

**Definition.** A local rule $F$ is a robust solution to pattern $T$ in timing model $S$ if, for all sizes $n$ and all configurations $X$ of size $n$ and all call sequences on $X$

$$\lim_{n \to \infty} F_s^n(X)$$

whenever $T$ contains one more instances of size $n$. 

**$G$** Underlying Geometry

**$T$** Configuration Space
Definition. A local rule $F$ is a robust solution to pattern $T$ in timing model $S$ if, for all sizes $n$ and all configurations $X$ of size $n$ and all call sequences on $X$

$$\lim_{n \to \infty} F_s^n(X)$$

whenever $T$ contains one more instances of size $n$. 

$G$ Underlying Geometry  

$S$ State set of size $m$ 

Configuration Space
The Model: Robust Solutions

**Definition.** A *local rule* $F$ is a robust solution to pattern $T$ in timing model $S$ if, for all sizes $n$ and all configurations $X$ of size $n$ and all call sequences on $X$

$$\lim_{n \to \infty} F^n_S(X)$$

whenever $T$ contains one more instances of size $n$.

- $G$ Underlying Geometry
- $S$ State set of size $m$
- $R$ Communication radius

Configuration Space

Thursday, November 28, 13
The Model: Robust Solutions

**Definition.** A local rule $F$ is a robust solution to pattern $T$ in timing model $S$ if, for all sizes $n$ and all configurations $X$ of size $n$ and all call sequences on $X$

$$\lim_{n \to \infty} F_s^n(X)$$

whenever $T$ contains one more instances of size $n$.

**Diagram:**

- **$G$** Underlying Geometry
- **$S$** State set of size $m$
- **$R$** Communication radius
- **$S$** Timing model

**Configuration Space**
The Model: Robust Solutions

**Definition.** A local rule $F$ is a robust solution to pattern $T$ in timing model $S$ if, for all sizes $n$ and all configurations $X$ of size $n$ and all call sequences on $X$

$$\lim_{n \to \infty} F^*_s^n(X)$$

whenever $T$ contains one more instances of size $n$.

**Diagram:**
- $G$: Underlying Geometry
- $S$: State set of size $m$
- $R$: Communication radius
- $T$: Timing model
- $T$: Pattern
- Configuration Space
The Model: Robust Solutions

**Definition.** A local rule $F$ is a robust solution to pattern $T$ in timing model $S$ if, for all sizes $n$ and all configurations $X$ of size $n$ and all call sequences on $X$

$$\lim_{n \to \infty} F_s^n(X)$$

whenever $T$ contains one more instances of size $n$.

---

Configuration Space

Disorder to Order

- $G$ Underlying Geometry
- $S$ State set of size $m$
- $R$ Communication radius
- $S$ Timing model
- $T$ Pattern

$$\implies F \text{ Robust Solution}$$
Other models: Amorphous Computing
Other models: Flocking & Sorting
Other models: Developmental Biology
Other models: Reconfigurable Robots
Other models: Pattern and Task Abstractions