NewVectors LLC
An Altarum Subsidiary

General Principles for Engineering Self-Organizing Applications for the Real World

Colloquium on Self-Organization
(DFG SPP 1183 Organic Computing)

Berlin-Adlershof 02/15/2007

Dr. Sven A. Brueckner
NewVectors, LLC
3520 Green Court, Suite 250
Ann Arbor, MI 48105, USA
p: +1.734.302-4683
e: sven.brueckner@newvectors.net
w: http://www.newvectors.net/staff/brueckners

Overview

- Introduction
- Definitions
- Simple Examples
  - "Natural" SO Systems
  - Engineered SO Systems
- Characteristics of SO Applications
- General Principles for Engineering SO Applications
  - Design
  - Analysis
- Conclusion

Introduction

- **1991-1997 Humboldt University Berlin**
  Institute for Computer Science (AI Lab)
- **1996-2000 DaimlerChrysler AG Research & Technology**
  Multi-Agent Systems Group
- **2000 Ph.D. at Humboldt University Berlin**
  "Return from the Ant – Synthetic Ecosystems for Manufacturing Control"
- **2000-now ERIM/Altarum/NewVectors**
  Agent-Based Complex Systems Group (PI / PM)
  - DARPA, DTO, NGA, ONR, NIST/ATP
    - Coordination of (Robotic) Vehicles
    - Prediction/Decision Support in Urban Combat
    - Improvised Explosive Devices (IED) Prediction
    - Information Management & Collaboration Support for Intelligence Analysts
    - Decision Support for Automotive Design and Launch

Overview

- Introduction
- **Definitions**
- Simple Examples
  - "Natural" SO Systems
  - Engineered SO Systems
- Characteristics of SO Applications
- General Principles for Engineering SO Applications
  - Design
  - Analysis
- Conclusion

Organization (1)

Organization$_1$ is a mapping from a system to an ordered set, e.g.,

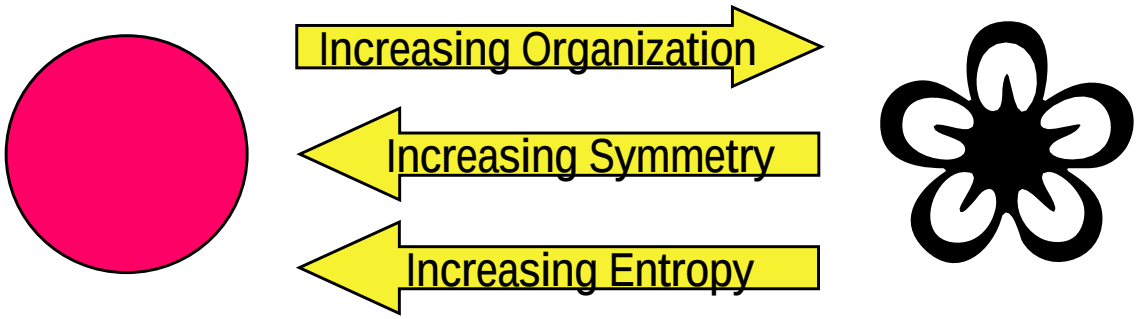
$$\text{Org: \{Systems\}} \rightarrow \Re^+$$

Different detailed definitions are possible. Common themes will include *entropy* S and *symmetry*.

Given two systems A = {a_1, …, a_n} and B = {b_1, …, b_k}, Org(A) > Org(B) if

- **S(B) > S(A) or**
- **B has a higher order of symmetry than A**

(Different bases for measuring entropy yield different kinds of organization: behavioral, state, …)



Organization (2)

Organization$_2$ is a *process* in a single system in which Organization$_1$ increases with time:

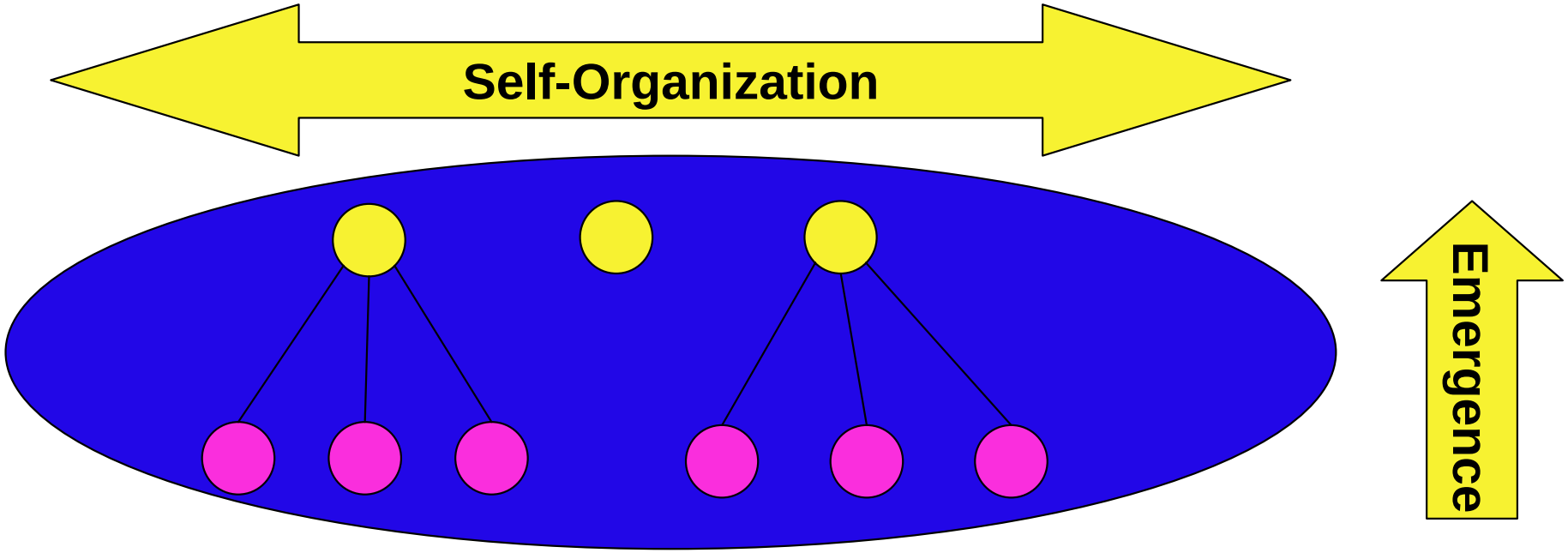
$$Org(A(t_2)) > Org(A(t_1)),\ t_2 > t_1$$

Organization$_3$ is the *structure* resulting from Organization$_2$, and can be measured with Organization$_1$.

Definitions

Self-Organization: Organization

- **Among elements *within a level***
- ***Without information flow* across the boundary**
- **Depends critically on definition of *system boundary***
- **A system could be partially self-organized**

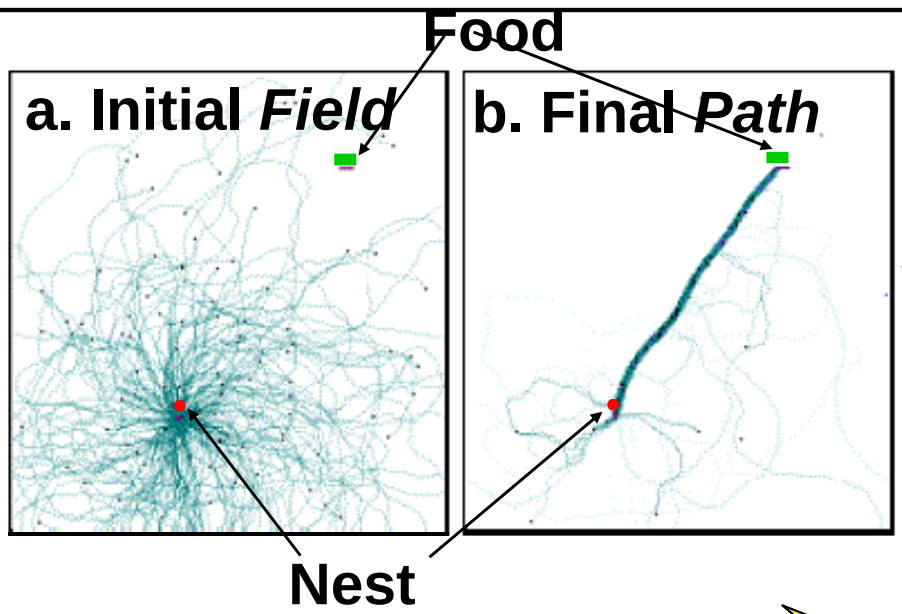


Emergence: Appearance of structures at a *higher level* that are not explicitly represented in *lower-level* components or external commands.

Overview

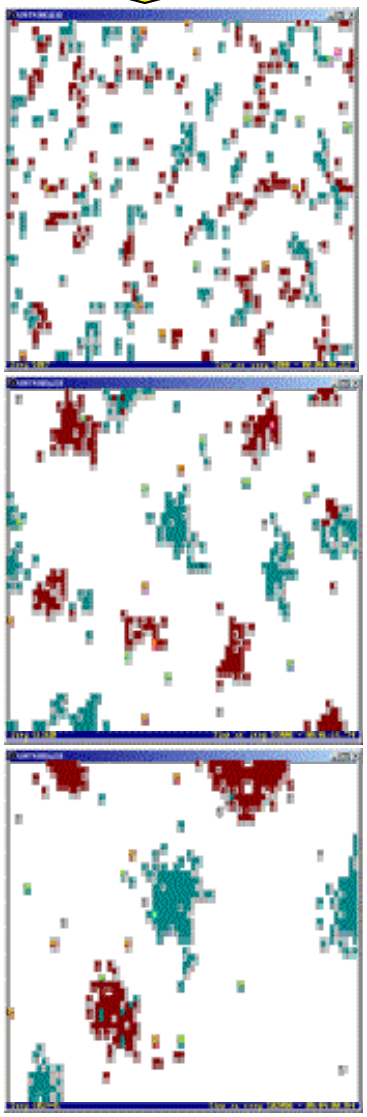
- Introduction
- Definitions
- **Simple Examples**
  - **“Natural” SO Systems**
  - **Engineered SO Systems**
- Characteristics of SO Applications
- General Principles for Engineering SO Applications
  - Design
  - Analysis
- Conclusion

Examples: "Natural" SO Systems

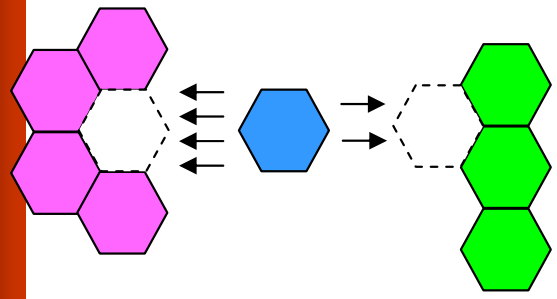


Ants: Path Formation (pheromones)

Ants: Nest Sorting (concen-tration)

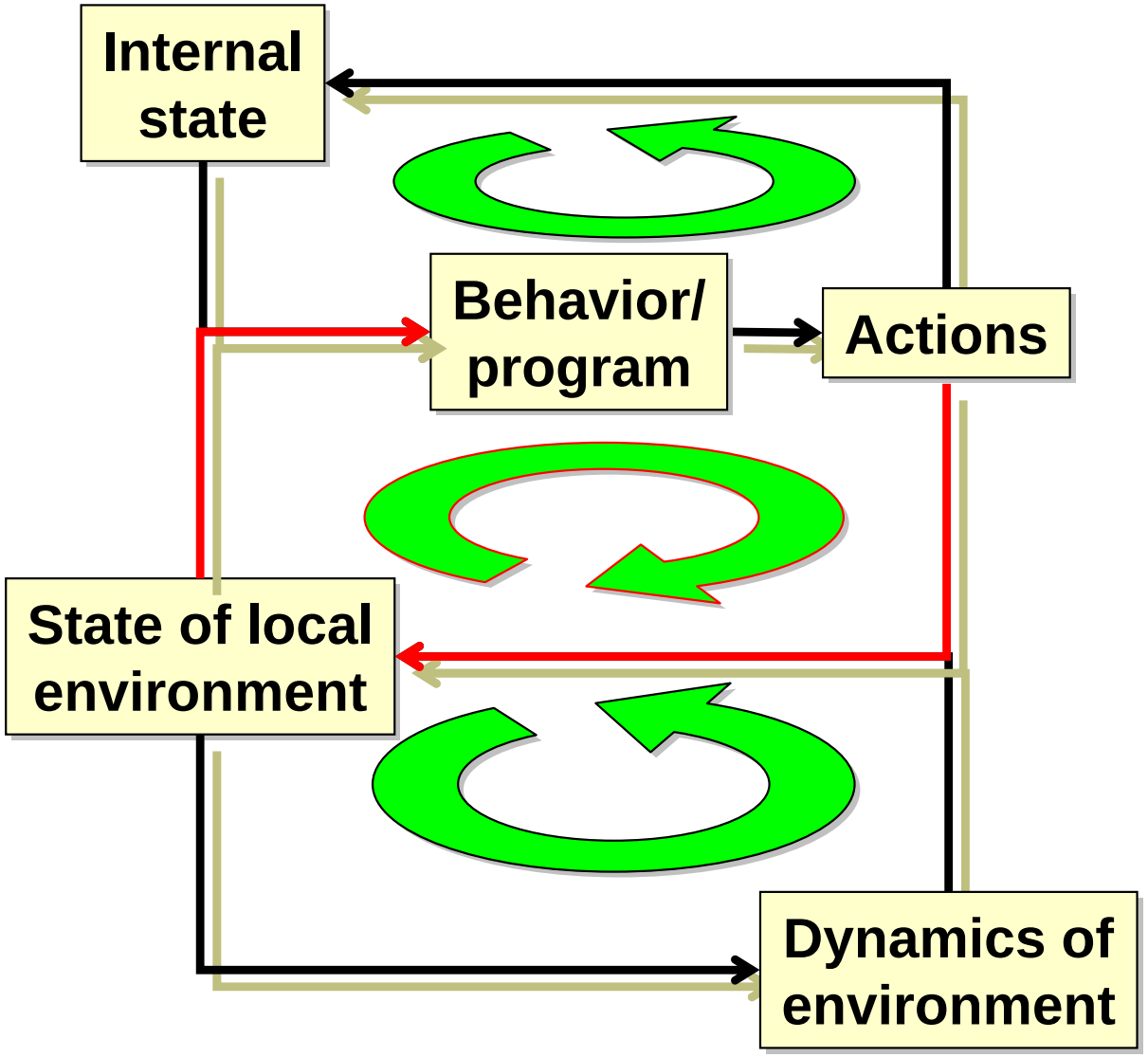


Common Principle: **Stigmergy**



Wasps: Nest Construction (shape fitting)

General Stigmergic Schema



στιγμα = "sign"

εργον = "action"

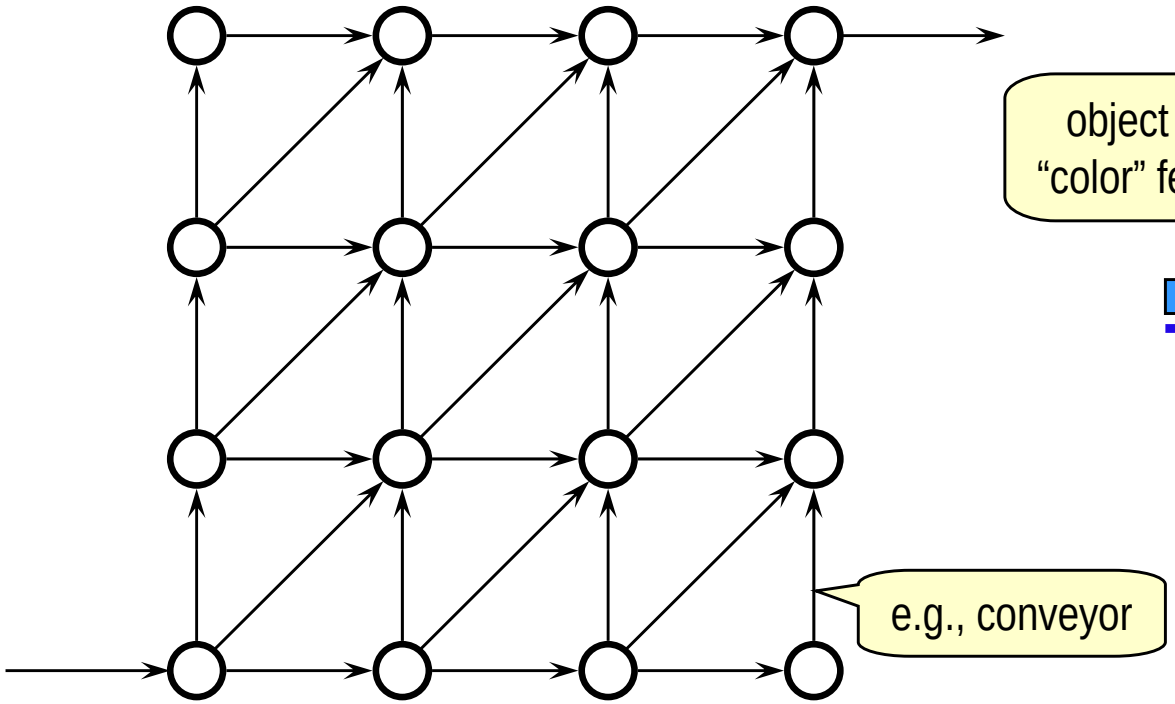
Actions →Signs in the environment → Actions → Signs → …

Environmental Topology → Locality of Interaction

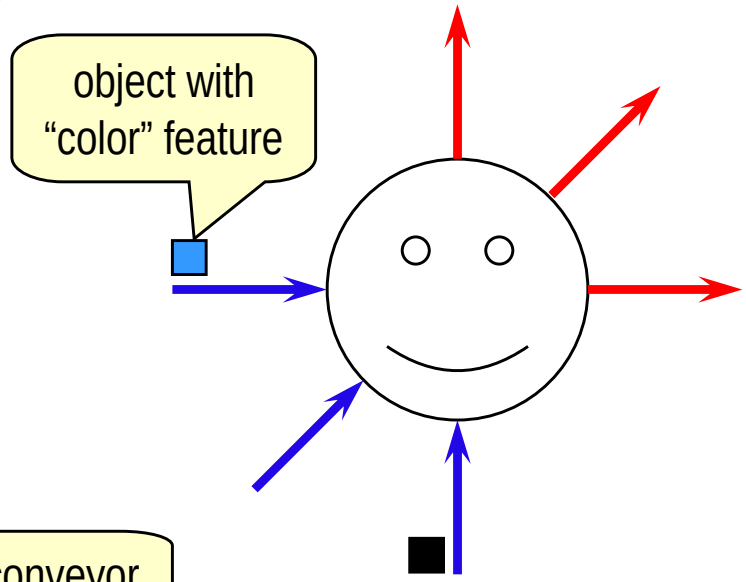
Possible topologies:

- Manifolds
- Lattices (semantic networks)
- The Internet
- …

Examples: Engineered SO Systems



(Material) Flow Topology



Decision Point
When to act?
What to do?

↓

<u>Prefer</u> to send objects of the same “color” as the previous one to an exit.

Examples of Human Self-Organization

Pre-Computer Examples

- Movement Coordination
  - Trail Formation
  - Traffic Flow
- Market Systems
- Elections
- Document Editing
- Boards
  - Bulletin
  - In/Out
  - Situation
- Viral Marketing

H. V. D. Parunak. A Survey of Environments and Mechanisms for Human-Human Stigmergy. In D. Weyns, F. Michel, and H. V. D. Parunak, Editors, *Proceedings of E4MAS 2005*, vol. LNAI 3830, *Lecture Notes on AI*, pages 163-186. Springer, 2006. www.newvectors.net/staff/parunakv/HumanHumanStigmergy2005.pdf

Computer Examples

- Intelligent Highway Systems
- Collaboration Environments
  - Content
    - Linking (WWW, Wiki, Groove, eMail List Servers)
    - Ranking (Google)
    - Sharing (Napster, Mobius, Grokster, …)
  - Process:
    - RAPPID (electromechanical design)
    - Coordinators (battle plan adjustment)
    - Workflow / BPM systems
    - Auction sites (eBay)
- Recommender Systems (Amazon)
- Resource Allocation
- Scheduling and Planning

Overview

- Introduction
- Definitions
- Simple Examples
  - "Natural" SO Systems
  - Engineered SO Systems
- **Characteristics of SO Applications**
- General Principles for Engineering SO Applications
  - Design
  - Analysis
- Conclusion

Characteristics of SO Applications

Application Characteristic	Motive
Discrete	Needed for Agents (and some kinds of Organization)
Diverse	Mixed capabilities may require on-the-fly role assignment
Deprived (resource constrained)	Rules out brute-force methods
Distributed	Defines topology within which Organization takes place
Decentralized	Avoids scaling problems
Dynamic in • Scope • Speed • Obscurity	Rules out pre-designed organization

Overview

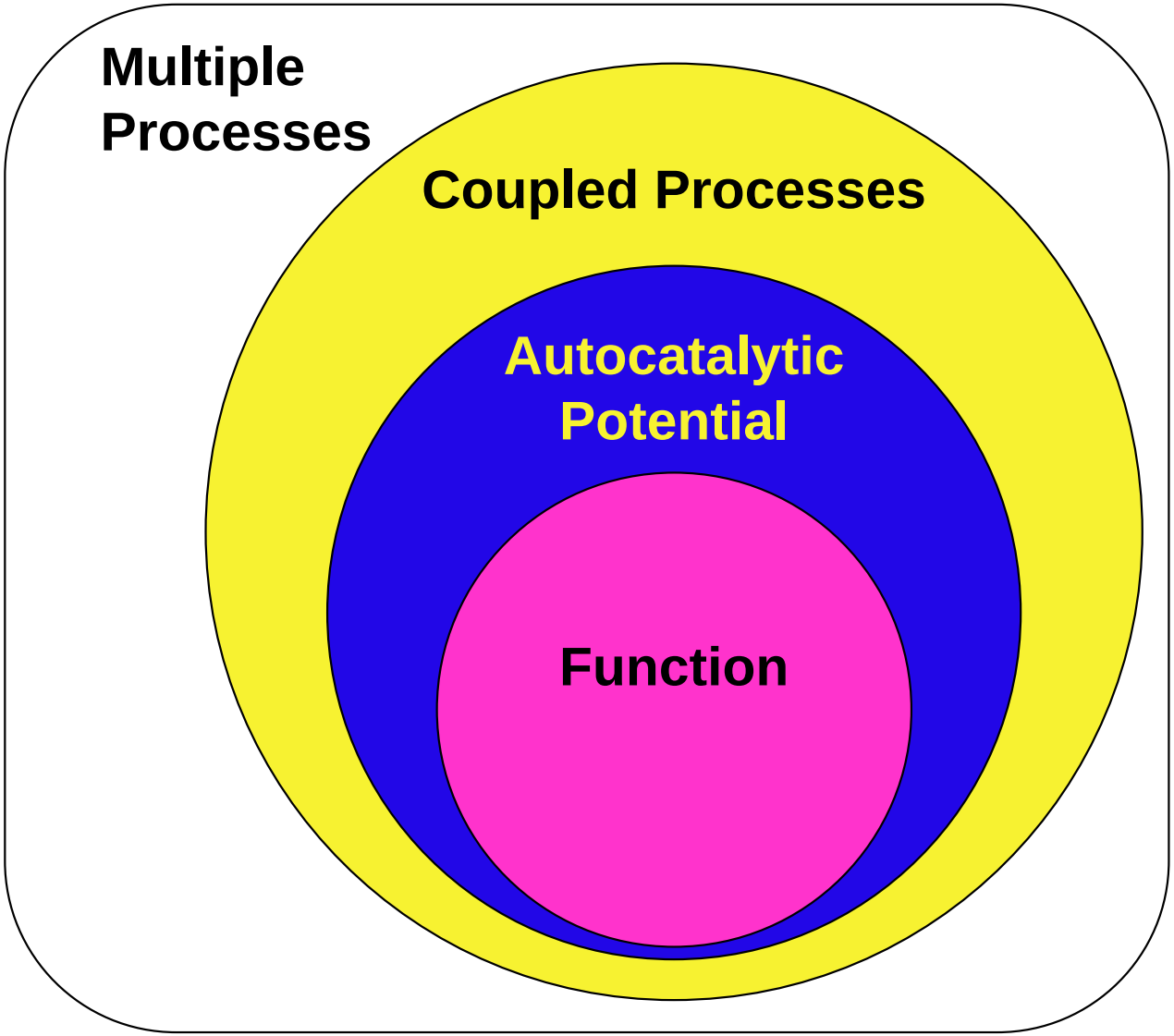
- Introduction
- Definitions
- Simple Examples
  - "Natural" SO Systems
  - Engineered SO Systems
- Characteristics of SO Applications
- **General Principles for Engineering SO Applications**
  - **Design**
  - **Analysis**
- Conclusion

Why Systems Self-Organize

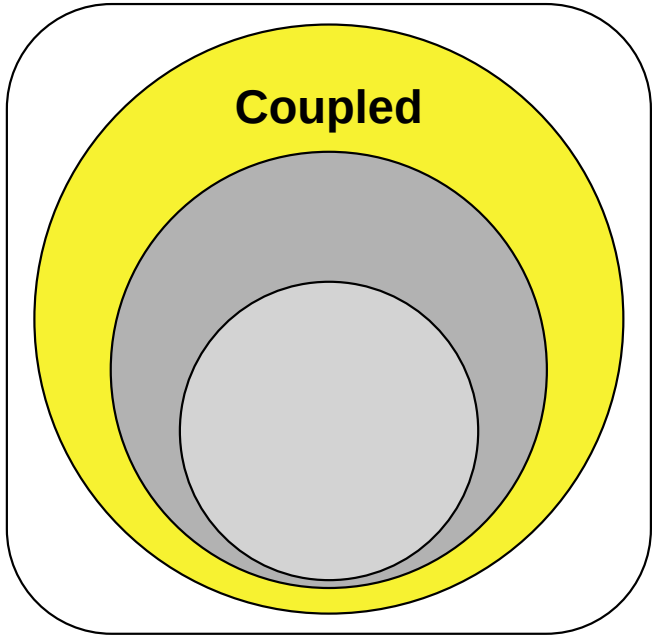
What enables a set of processes (agents) to organize themselves?

Consider three restrictions of the space of all such sets of processes.

Derive principles from each restriction.



Design Principles from Coupling



Basic system structure:

- **Multiple agents**
- **Shared environment for sensing and acting**
- **Contingent behavior**

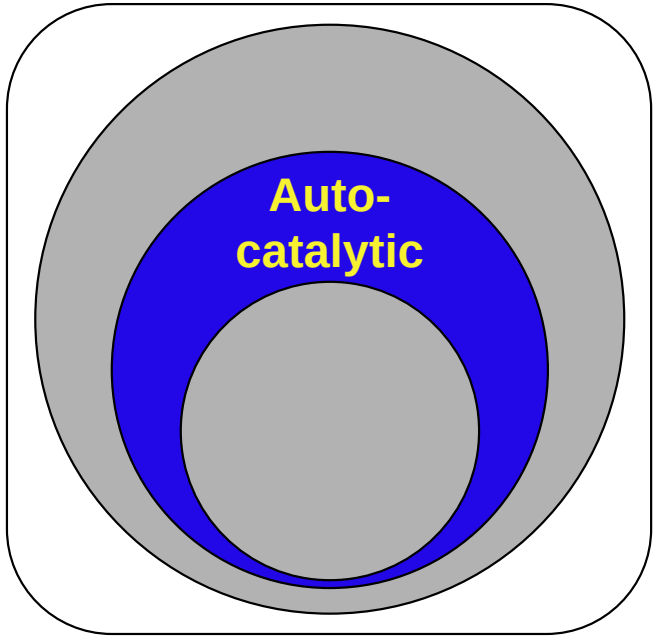
Scalability requires Local Interaction, which requires agents to

- **Be small**
- **Represent entities (which tend to be local) rather than functions (which tend to be global)**
- **Decentralize control**

Keep Agents Small

- in Mass: Small compared with the whole system ➔ agents are numerous
  - Example: One termite (0.1 gram) vs. hill (10 tons)
  - Inexpensive to implement
  - Easy to understand
  - Limited impact of unit failure
  - Support in industrial controls marketplace (Rockwell AB, Mitsubishi, ZWorld, Echelon, ...)
- in Time: Bounded (forgetful)
  - Example: Pheromone evaporation
  - Able to adapt to environmental changes
- in Space: Local sensing & action

Design Principles from Autocatalyticity



A set of agents has *autocatalytic potential* if in some regions of their joint state space, their interaction causes system entropy to decrease.

In that region of state space, they are *autocatalytic*.

Even better if *autocatalytic* region is in the basin of an *attractor*.

Establish Feedbacks among system elements (natural for stigmergy)

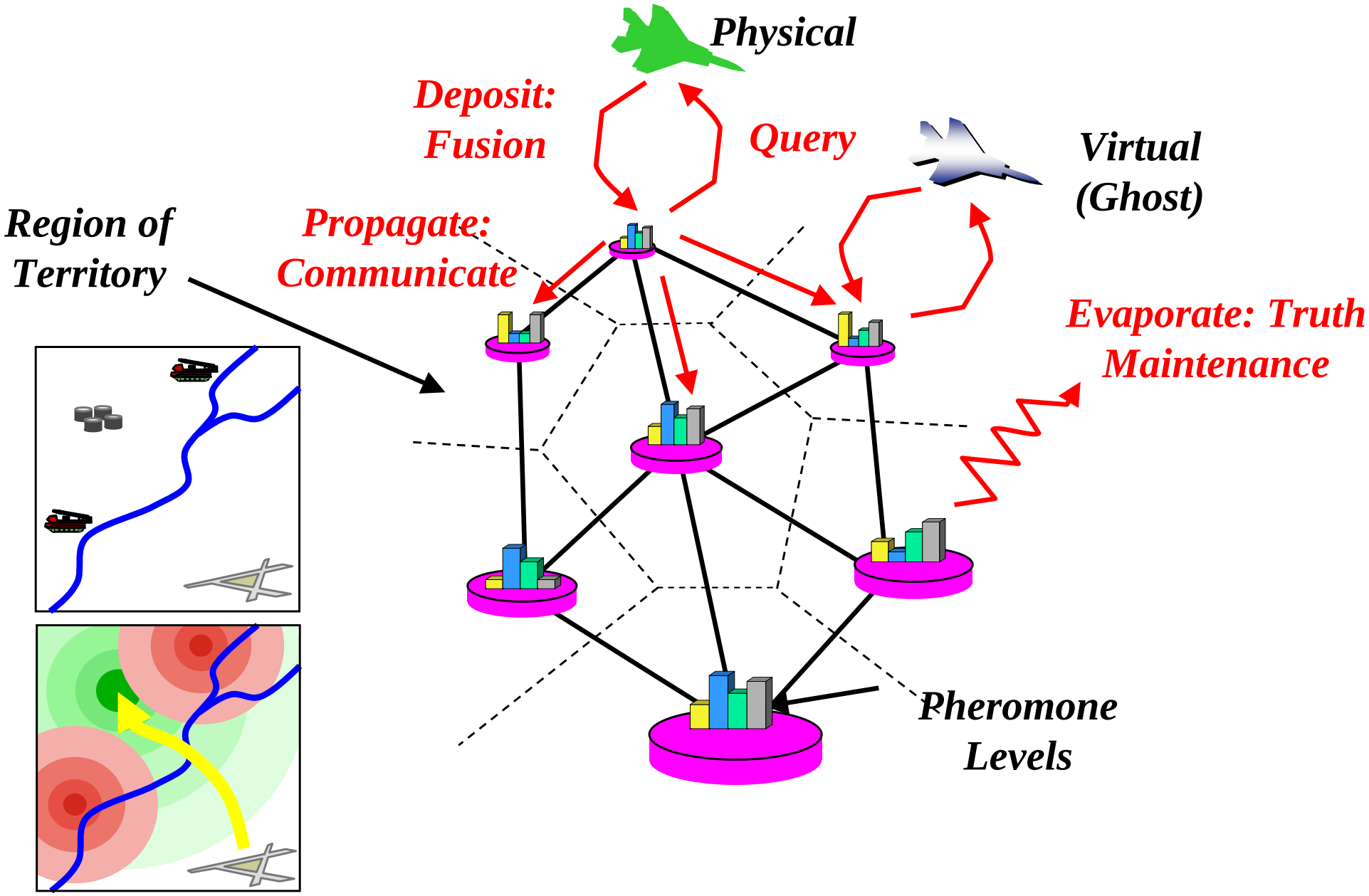
- Positive feedback to keep from dying out
- Negative feedback to avoid explosion

Think Flows rather than Transitions

Diversify agents to keep flows going

- Stochasticity: Random element in decisions
- Situation in the environment → movement
- Stateful action (learning/forgetting)

Flows in Digital Pheromones
(Parunak et al. 2004)



Support Agent Diversity: How?

1. Speciation

- Multiple agent types (costly to implement)

2. Repulsion

- Draw on physical exclusion laws

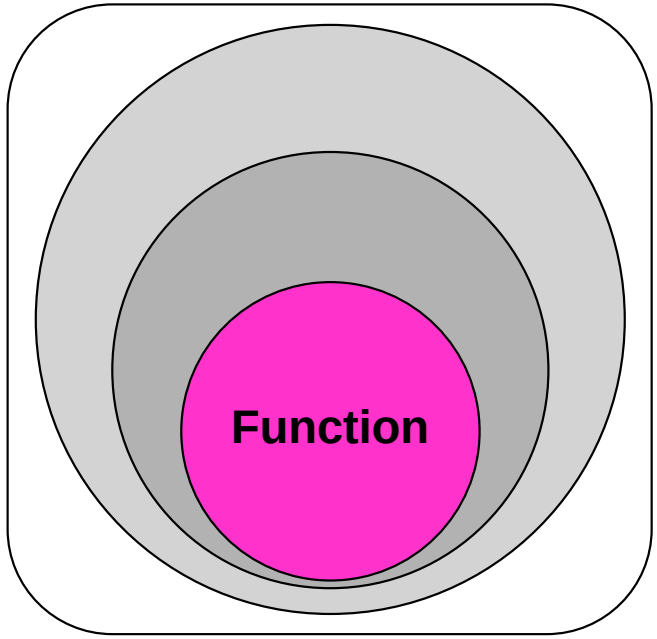
3. Randomness

- Less dependent on explicit model of the domain
- Simple processing
- Wasps: probability of winning face-off $1/(1+e^{h(F1-F2)})$
- Ants: probability of pick-up $k/(k+f)$

4. Differential rewards

- Adaptive mechanism in agent to adjust behavior

Design Principles for Functionality



Functional Diversity → selective Potential

- Each agent supports multiple functions
- Each function requires multiple agents
- Break symmetry among agents to explore behavior space
- Functionality emerges from agent interactions

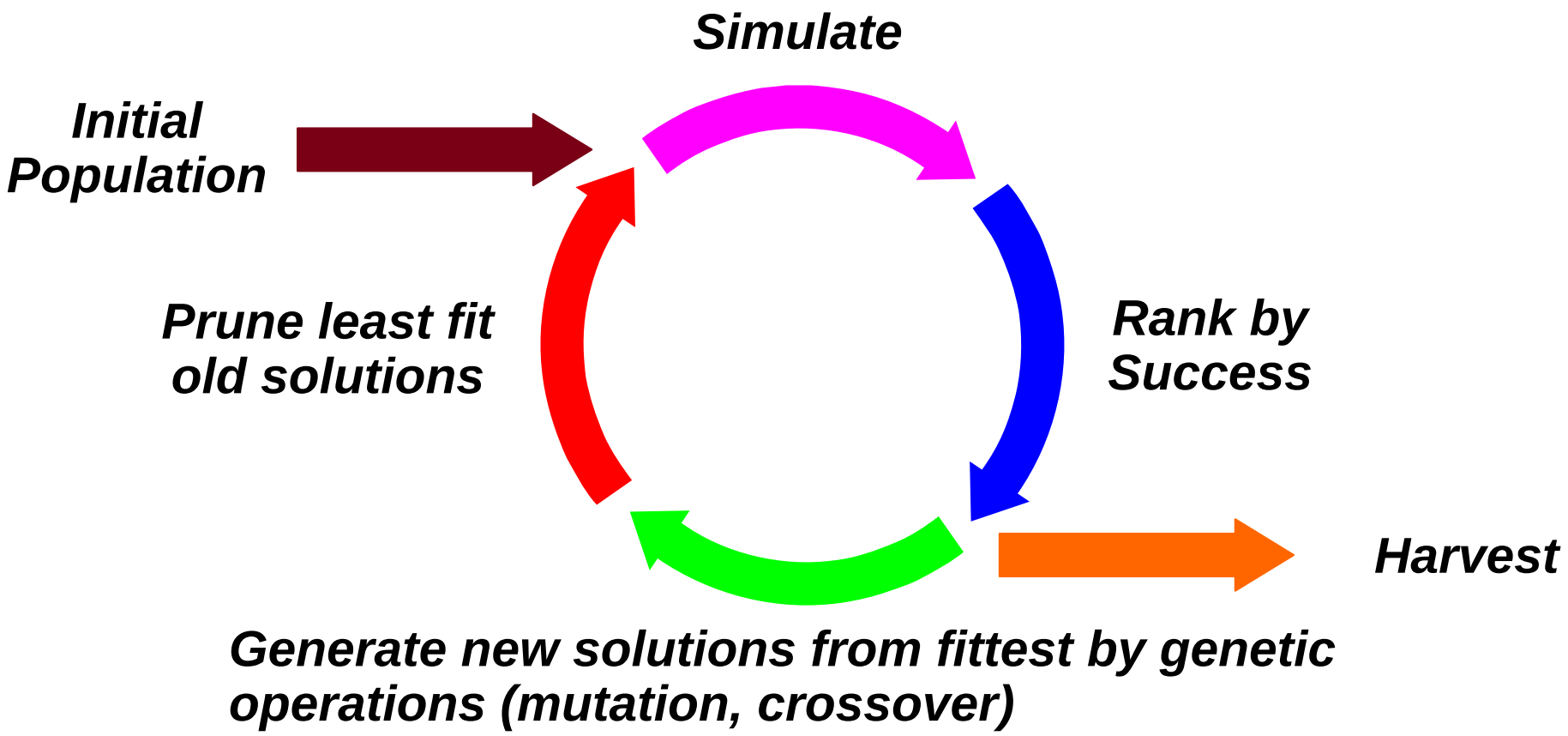
Selection Pressure: requires fitness measure

- Ideal: locally visible measure that estimates global goodness of solution
- Example: agent's Option Set Entropy as measure of system convergence

Selection Mechanisms

- Genetic: *composition* of population varies
- Particle Swarm: *behaviors* of individuals vary

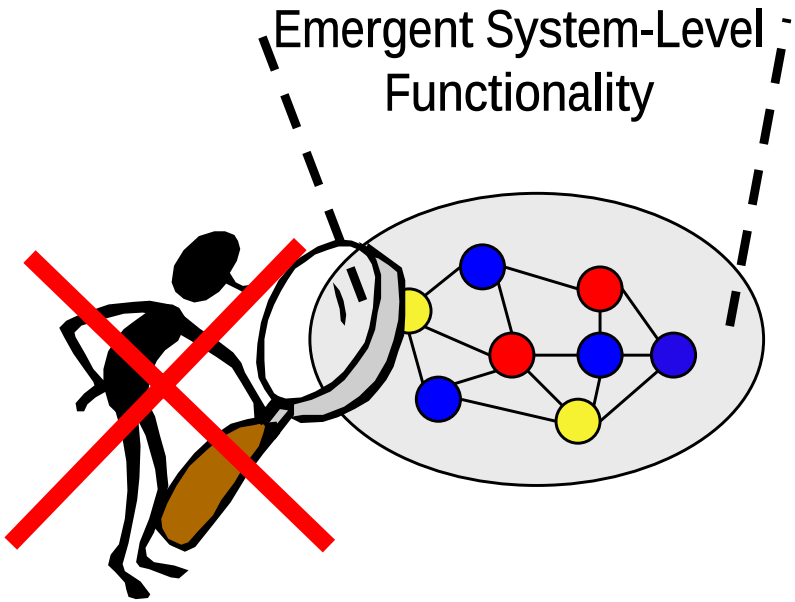
Synthetic Evolution

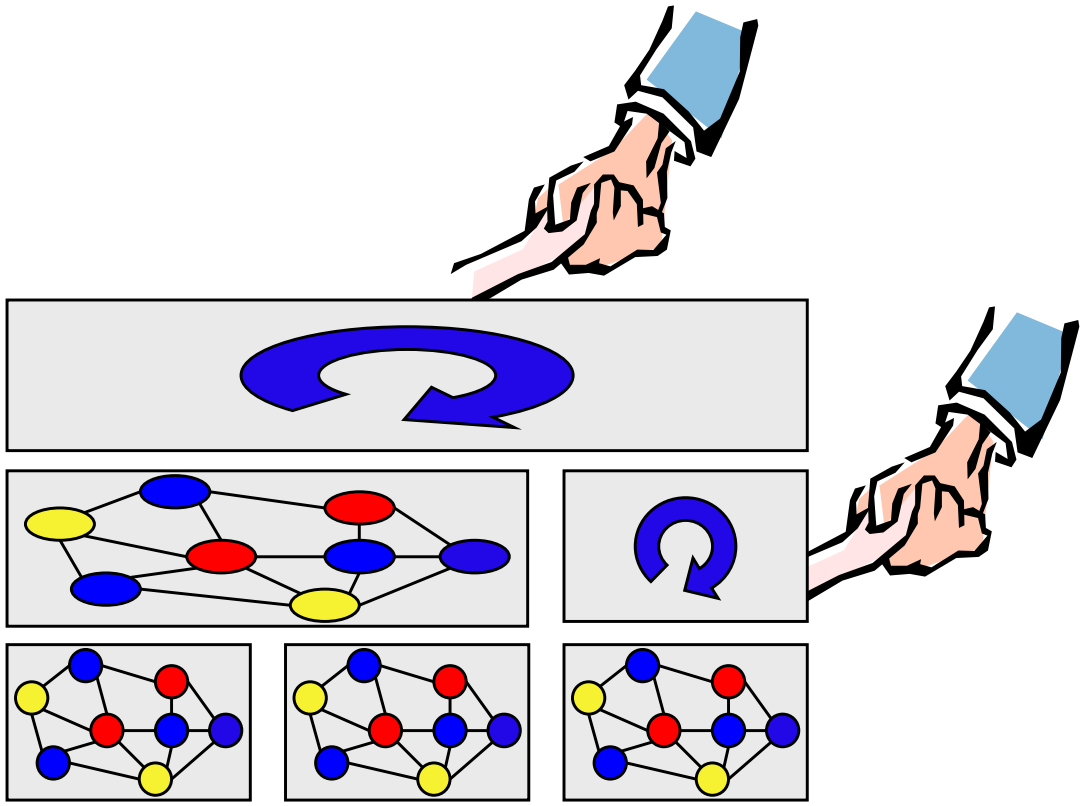


Jacob, 2001

Self-Organizing Functionality, not Application

Humans have a hard time interacting with self-organizing systems at the individual component level.

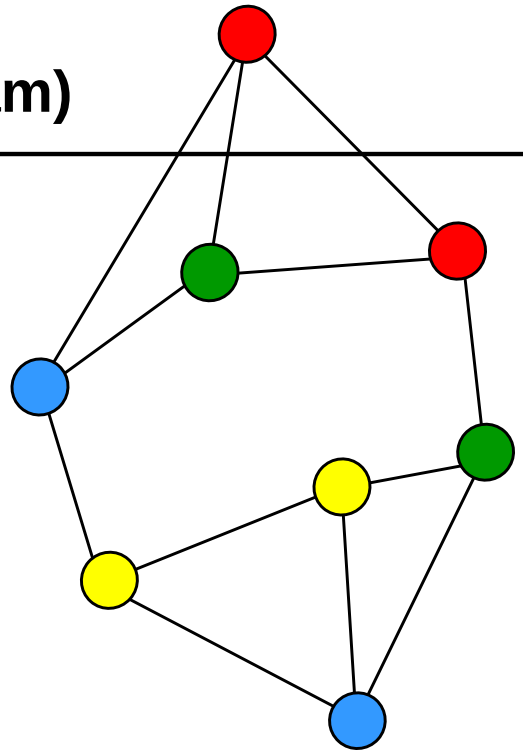




Overview

- Introduction
- Definitions
- Simple Examples
  - "Natural" SO Systems
  - Engineered SO Systems
- Characteristics of SO Applications
- **General Principles for Engineering SO Applications**
  - Design
  - **Analysis**
    - **Phase Structure**
    - **Systematic Exploration through Simulations**
- Conclusion

Example
(from DARPA ANT Program)



distributed graph coloring algorithm
(Stephen Fitzpatrick, Kestrel Institute
(model of resource allocation)

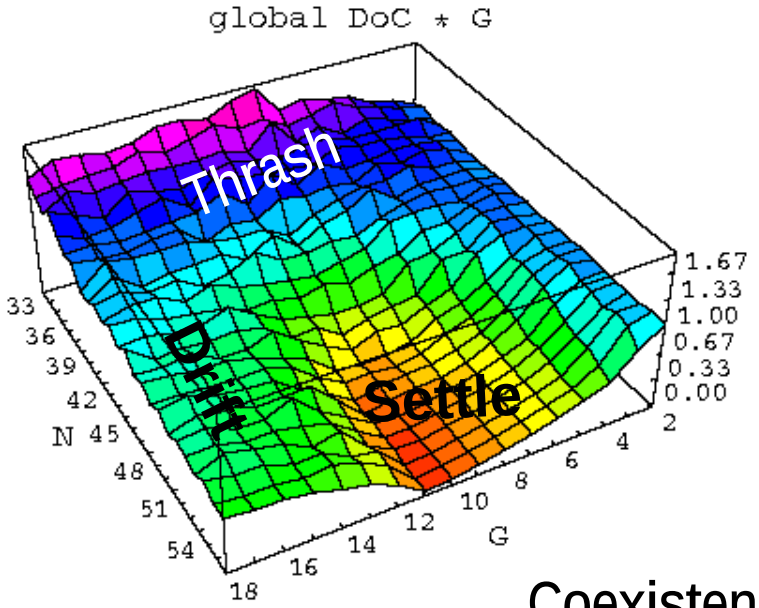
- Agent = Node
  - Awaken with probability AL
  - Compare own color with current knowledge of neighbor color
  - Adjust own color stochastically
  - Communicate new color to neighbors with latency CL

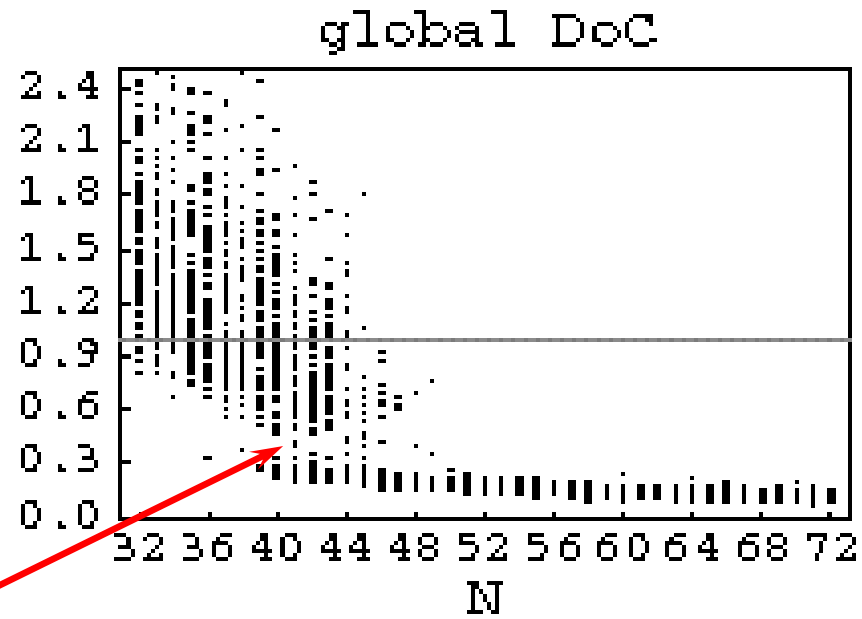
Performance Metrics

- Degree of Conflict (average over nodes)
- Option Set Entropy
- False Information Percentage

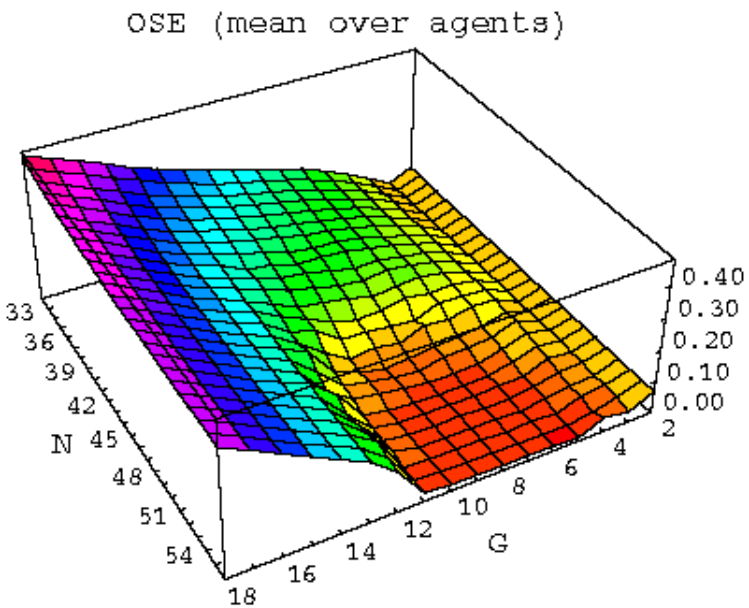
Model Parameter	Description
N	Number of Nodes in Graph
K	Number of Neighbors per Node
G	Number of Colors Available
GC	Mechanism to Construct Random Graph
AL	Probability to Activate Color Decision Process
RP	Probability to Randomly Change Color
CL	Time of Message Transfer to Nearest Neighbor
MD	Constraint on Permitted Change of Local DoC
CS	Mechanism to Select Color from Option Set

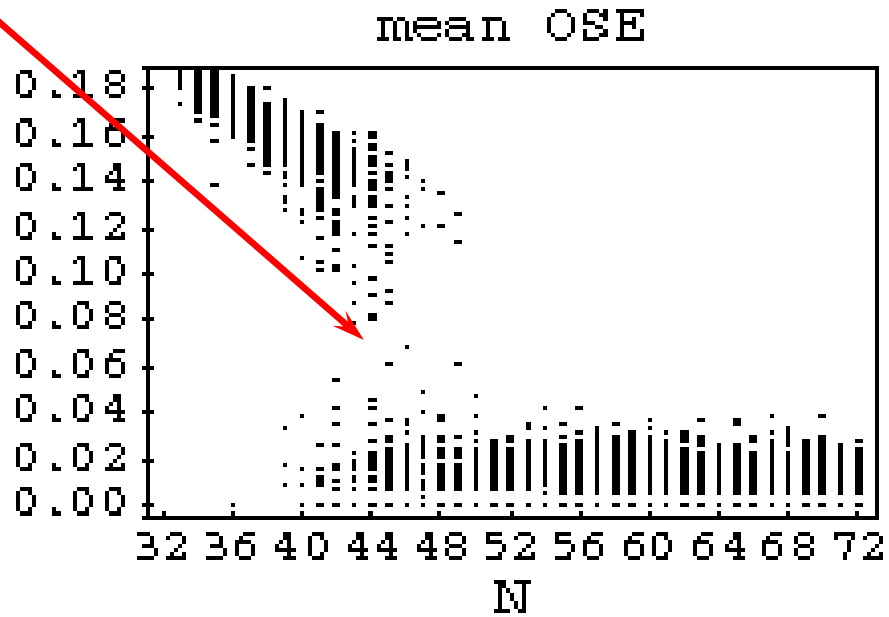
Emergent Dynamics



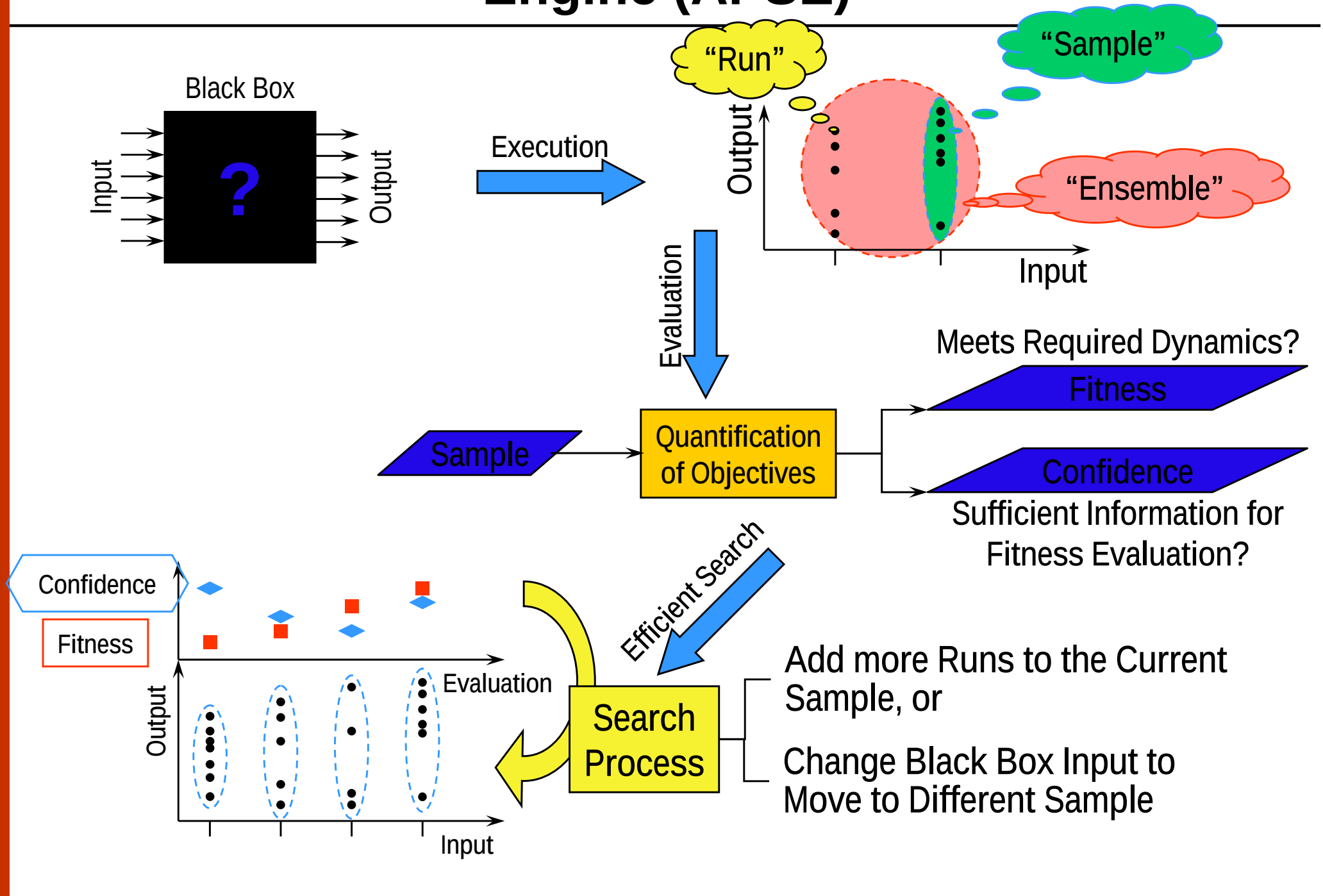


Coexistence Region

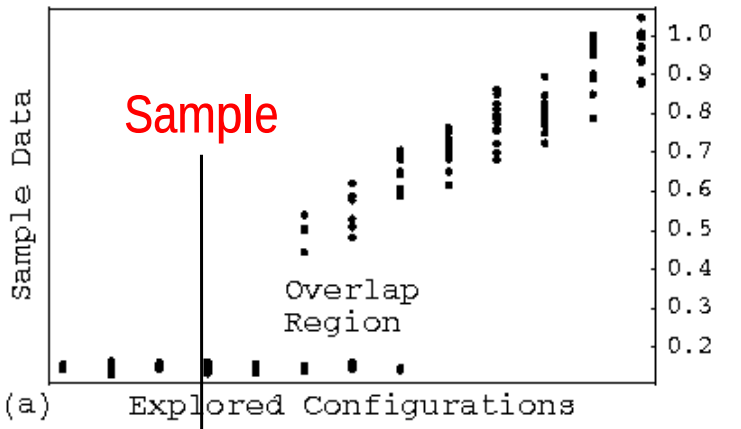




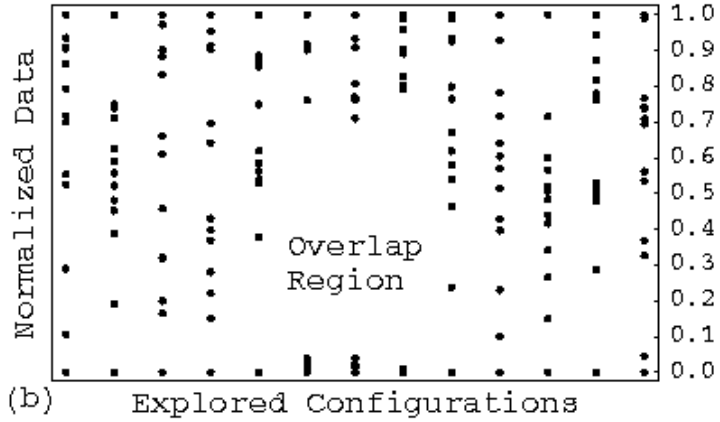
Adaptive Parameter Search Engine (APSE)

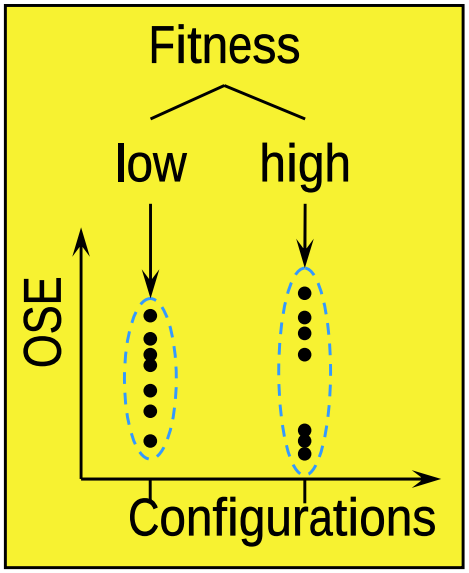


Example
(from DARPA ANTs Program)

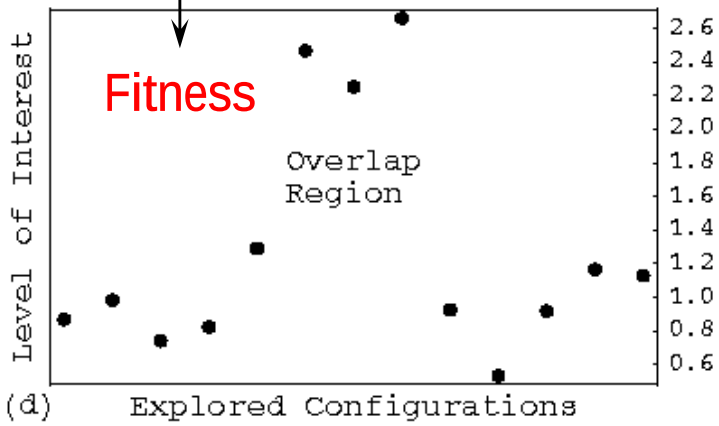


stretch data
btw. 0 and 1

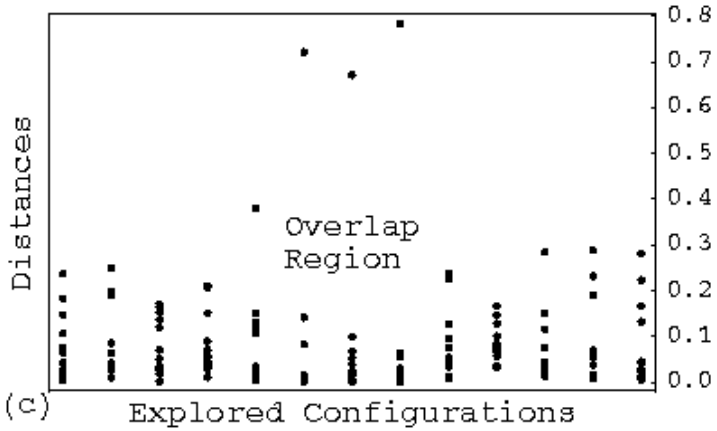




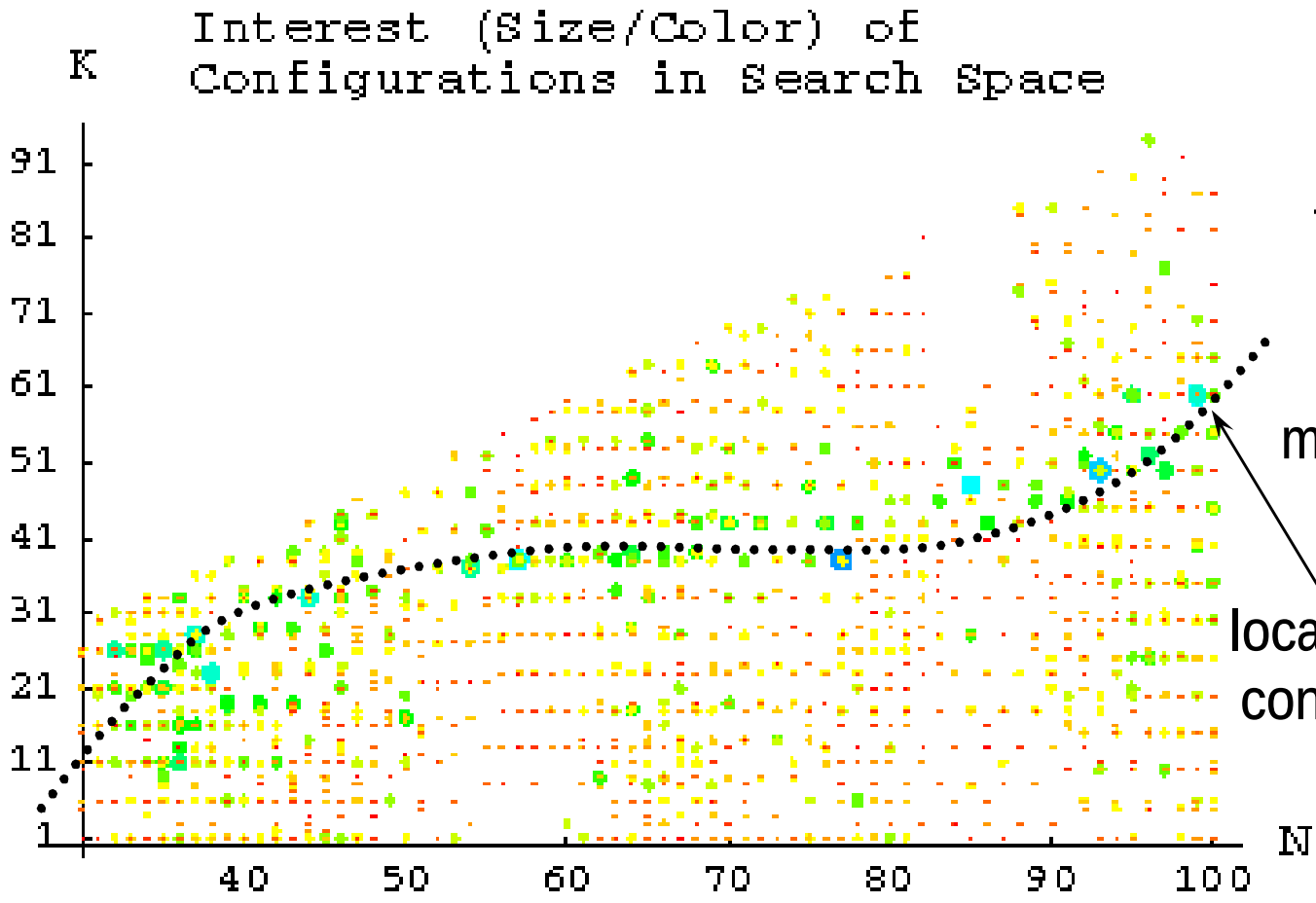
calculate distances
btw. neighboring points



CoVar
across
distances



Example
(from DARPA ANTs Program)



small red
few runs & low fitness

large green
many runs & high fitness

location of overlap region confirmed by systematic sweep

Prototypical (ad hoc) Implementation
6 x reduction in simulation effort to find phase change

Overview

- Introduction
- Definitions
- Simple Examples
  - "Natural" SO Systems
  - Engineered SO Systems
- Characteristics of SO Applications
- General Principles for Engineering SO Applications
  - Design
  - Analysis
- **Conclusion**

Conclusions

- Rigorous definitions of SO concepts motivate and guide application design, analysis and deployment process
- Fine-grained multi-agent approach yields intended complex and adaptive system-level behavior
- Stigmergic agent coordination is fundamental principle to harness power of SO
- SO applications have high-dimensional state space riddled with phase changes and attractors
  - Design: shape the state space to guide the system to intended behavior
  - Analysis: statistical physics techniques + extensive simulations

Thank You for Your Attention!