Self-Organization
Swarm Intelligence

Winter Semester 2010/11
Motivation for Self-Organization

- Problem of today’s networks
  - Heterogeneity
  - Dynamics
  - Scalability

- Method: Self-organization
  - Fast, autonomous reaction to problems
  - Automation of control
  - Distributed control

- Some application scenarios:
  - Severe network impact due to disasters
  - Energy savings
  - Privately operated femto cells
Outline

• Properties
• Self-Organization and Emergence
• Basis Methods
• Swarm Intelligence
  – Principles
  – Communication Forms
  – Stigmergy
  – Ant Colony Optimization (ACO)
• Conclusion
• References
### Self-Organization – Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No central control</td>
<td>No global control system</td>
</tr>
<tr>
<td></td>
<td>No global information</td>
</tr>
<tr>
<td></td>
<td>Subsystems perform completely autonomous</td>
</tr>
<tr>
<td>Emerging structures</td>
<td>Global behavior or functioning of the system emerges in form of observable pattern or structures</td>
</tr>
<tr>
<td>Resulting complexity</td>
<td>Even if the individual subsystems can be simple as well as their basic rules, the resulting overall system becomes complex and often unpredictable</td>
</tr>
<tr>
<td>High scalability</td>
<td>No performance degradation if more subsystems are added to the system</td>
</tr>
<tr>
<td></td>
<td>System performs as requested regardless of the number of subsystems</td>
</tr>
</tbody>
</table>
Self-Organization and Emergence

Definition **Self-Organization**
Self-organization is a process in which structure and functionality (pattern) at the global level of a system emerge solely from numerous interactions among the lower-level components of a system without any external or centralized control. The system's components interact in a local context either by means of direct communication or environmental observations without reference to the global pattern.

Definition **Emergence**
Emergent behavior of a system is provided by the apparently meaningful collaboration of components (individuals) in order to show capabilities of the overall system (far) beyond the capabilities of the single components.
Basis Methods in Self-Organizing Systems

- Positive and negative feedback
- Interactions among individuals and with the environment
- Probabilistic techniques
Positive and Negative Feedback

Simple feedback

Feedback

Input → System state → Output

Measurement

Amplification problems

Snowballing effect

Implosion effect
Interactions among Individuals and with the Environment

- Direct communication among neighboring systems
- Indirect communication via the environment (stigmergy)
- Interaction with (stimulation by) the environment
Probabilistic Techniques

- Examples: stochastic processes, random walk
- Objectives: leaving local optima, stabilization

Nest-building algorithm used by termites to collect wood chips

```
Ensure: Collection of wood chips on a single heap
1: initialize
2: loop
3: repeat
4: random walk
5: until found wood chip
6: if already carrying a wood chip then
7: drop wood chip
8: else
9: take wood chip
10: end if
11: end loop
```
Swarm Intelligence
Swarm Intelligence (SI)

• Swarm Intelligence (SI) is
  
  – A computational technique for solving distributed problems inspired from biological examples provided by
    • social insects such as ants, termites, bees, and wasps and by swarm, herd, flock, and
    • shoal phenomena such as fish shoals
  
  – An approach to controlling and optimizing distributed systems
  
  – Resilient, decentralized, self-organized technique
SI Organizing Principles

SI has the following notable features:

• Autonomy:
  ➔ The system does not require outside management or maintenance. Individuals are autonomous, controlling their own behavior both at the detector and effector levels in a self-organized way.

• Adaptability:
  ➔ Interactions between individuals can arise through direct or indirect communication.

• Scalability:
  ➔ SI abilities can be performed using groups consisting of a few, up to thousands of individuals with the same control architecture.

• Flexibility:
  ➔ No single individual of the swarm is essential, that is, any individual can be dynamically added, removed, or replaced.
SI Organizing Principles

SI has the following notable features:

• Robustness:
  ➔ No central coordination takes place, which means that there is no single point of failure

• Massively parallel:
  ➔ Tasks performed by each individual within its group are the same

• Self-organization:
  ➔ The intelligence exhibited is not present in the individuals, but rather emerges somehow out of the entire swarm.
SI Communication Forms

Indirect Communication
• Implicit communication that takes place between individuals via the surrounding environment.
• Known as Stigmergy communication.

Direct Communication
• Explicit communication that can also take place between individuals.
• Examples:
  • waggle dance of the honeybee,
  • trophallaxis (food or liquid exchange, e.g., mouth-to-mouth food exchange in honeybees),
  • …
Stigmergy

- Stigmergy: stigma (sting) + ergon (work) → “stimulation by work”

- Characteristics of stigmergy
  - Indirect agent interaction modification of the environment
  - The information is local: it can only be accessed by insects that visit the locus in which it was released
  - Work can be continued by any individual

\[ S_I : \text{Pillar construction state} \]
\[ R_I : \text{Response} \]
Ant Colony Optimization (ACO)
Ants

• Why are ants interesting?
  – Ants solve complex tasks by simple local means
  – Ants productivity is better than the sum of their single activities
  – Ants are grand masters in search and exploitation
Foraging behavior of Ants (Double bridge experiment)

Ants start with equal probability
The ant on shorter path has a shorter time from it’s nest to the food.
Foraging behavior of Ants (Double bridge experiment)

The density of pheromone on the shorter path is higher.
The next ant takes the shorter route
Foraging behavior of Ants (Double bridge experiment)

After some time, the shorter path is almost exclusively used
Ant Colony Optimization (ACO)

- Developed by Dorigo and Di Caro
- It is a population-based metaheuristic used to find approximate solutions to difficult optimization problems
- ACO is structured into three main functions:
  1. AntSolutionsConstruct( )
     - Performs the solution construction process
  2. PheromoneUpdate( )
     - Performs pheromone trail updates
     - Includes also pheromone trail evaporation
  3. DaemonActions( )
     - An optional step in the algorithm which involves applying additional updates from a global perspective
ACO metaheuristic

- CO problem
- Pheromone model
- Solution components
- Probabilistic solution construction
- Pheromone value update
- Initialization of pheromone values
Properties of the artificial ant

- Each artificial ant has an internal memory

- Starting in an initial state $S_{\text{initial}}$, each ant tries to build a feasible solution to the given problem, moving in an iterative fashion through its search space.

- The guidance factors for ants' movement take is a transition rule which is applied before every move from state $S_i$ to state $S_j$.

- The amount of pheromone each ant deposits is governed by a problem-specific pheromone update rule.

- Pheromone deposition may occur at every state transition during the solution construction (pheromone trial update).

- Ants may retrace their paths once a solution has been constructed and only then deposit pheromone, all along their individual paths.
The original Ant System

• Developed by Dorigo et al. (1996)

• Ant system (AS)
  – First ACO algorithm
  – Pheromone updated by all ants in the iteration

• Travelling Salesman Problem (TSP) was used as a test-bed for this algorithm
Travelling Salesman Problem (TSP)

- TSP description:
  - Visit cities in order to make sales
  - Save on travel costs
  - Visit each city once (Hamiltonian circuit)

- A Hamiltonian cycle (or Hamiltonian circuit) is a cycle in an undirected graph which visits each vertex exactly once and also returns to the starting vertex.
Solution for TSP

• A connected graph $G = (V, E)$, where
  – $V$ is a set of vertices (cities)
  – $E$ is a set of edges (connection between cities)

• A variable called pheromone is associated with each edge and can be read and modified by ants

• Ant system is an iterative algorithm at each iteration,
  – A number of artificial ants are considered
  – Each ant build a solution by walking from vertex to vertex
  – Each vertex is visited one time only
  – An ant selects the following vertex to be visited according to a stochastic mechanism that is biased by the pheromone

• At the end, the pheromone values are updated on order to bias ants in the future iteration to construct solutions similar to the previously constructed
1. Pheromone trail
   - Iteration is defined as the interval in \((t,t+1)\) where each of the \(N\) ants moves once
   - Epoch \(\Rightarrow\) \(n\) iterations (when each ant has completed a tour)
   - Intensity of trail: \(\tau_{ij}(t)\)
   - Trail update function after each epoch:
     \[
     \tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^{N} \Delta \tau_{ij}^k
     \]
   - \(\rho\) is the evaporation rate
   - \(\Delta \tau_{ij}^k\) is the quantity of pheromone laid on path \((i,j)\) by the ant \(k\) and is given by:
     \[
     \Delta \tau_{ij}^k = \begin{cases} Q / L_k & \text{If ant } k \text{ used edge } (i,j) \text{ in its tour,} \\ 0 & \text{otherwise,} \end{cases}
     \]
     where \(Q\) is a constant and \(L_k\) is the tour length of \(k\)th ant.
Ant System and the TSP

2. Memory
   – Prevents town repeats
   – Tabu list

3. Awareness of environment
   – City distance
   – Visibility: \( \eta_{ij} = \frac{1}{d_{ij}} \)
     where \( d_{ij} = \sqrt{\left(x_i - x_j \right)^2 + \left(y_i - y_j \right)^2} \)

4. Probability function
   \[
   p_{ij}^k(t) = \begin{cases} 
   \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{j \in \text{allowed}} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta} & \text{if } j \in \text{allowed} \\ 
   0 & \text{otherwise} \end{cases}
   \]

\( \tau_{ij}(t) \) and \( \eta_{ij} \) are parameters influencing the pheromone and visibility respectively.
Some applications of ACO

• Scheduling

• Routing problems
  – Traveling Salesman Problem (TSP)
  – Vehicle routing
  – Network routing

• …
## Ant Foraging and ACO

<table>
<thead>
<tr>
<th>Biology (Ant Foraging)</th>
<th>ACO Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ant</td>
<td>Individual (agent) used to build (construct) a solution</td>
</tr>
<tr>
<td>Ant Colony</td>
<td>Population (colony) of cooperating individuals</td>
</tr>
<tr>
<td>Pheromone Trail</td>
<td>Modification of the environment caused by the artificial ants in order to provide an indirect mean of communication with other ants of the colony. Allows assessment of the quality of a given edge on a graph</td>
</tr>
<tr>
<td>Pheromone Evaporation</td>
<td>Reduction in the pheromone level of a given path due to aging</td>
</tr>
</tbody>
</table>
Conclusion

Self-organization: what does it mean?
  – distributed, autonomous control of systems
  – simple local algorithms or behavior
  – interaction with neighbors and/or environment only

Lots of applications of self-organization already
  – self-organization keeps today’s Internet together
  – development and tuning of self-organized algorithms takes time
  – stability is an issue

Swarm Intelligence
  – is a rich source of inspiration for our computer systems.
  – has many features that are desirable for distributed computing such as auto-configuration, auto-organization, autonomy, scalability, flexibility, robustness, emergent behavior, and adaptability
References

Self-organization in general


Application of Self-organization to communication networks


Swarm Intelligence

Integrated Communication Systems Group
Ilmenau University of Technology

Univ.-Prof. Dr.-Ing. Andreas Mitschele-Thiel
fon: +49 (0)3677 69 2819
fax: +49 (0)3677 69 1226
e-mail: mitsch@tu-ilmenau.de

Visitors address:
Technische Universität Ilmenau
Gustav-Kirchhoff-Str. 1
(Informatikgebäude, Room 210)
D-98693 Ilmenau

www.tu-ilmenau.de/ics
Der FOSSLC e.V. startet mit einem weiteren spannenden Workshop ins neue Jahr.
Ihr wolltet schon immer mal wissen, wie es bei Linux unter der Haube aussieht?
Wir zeigen es euch und bauen zusammen unseren eigenen Echtzeit-Kernel!

Auch dieses Mal könnt ihr euren PGP Schlüssel signieren lassen und euch CAcert Punkte holen!

Teilnehmerzahl begrenzt!

26.
19. Januar 2011 - 16:30 - 20:00 Uhr
TU Ilmenau | UniRZ RTK
Anmeldung unter event@fosslc.de

www.fosslc.de