Swarm Intelligence
(Ant Colony Optimization)

Prof. Dr.-Ing. Habil Andreas Mitschele-Thiel
M.Sc.-Inf Mohamed Kalil
Course description

Introduction
- Course overview
- Concepts of System Engineering

Mechanisms
- Swarm Intelligence
- Distributed AI
- Data Mining
- Neural Networks
- Reinforcement Learning

Application
- MANET
- Sensor Networks
- Information Management
- LTE
- Service Placement
- Cognitive Networks

Methodology

Prof. Dr.-Ing. habil. Andreas Mitschele-Thiel
Integrated HW/SW Systems Group

Swarm Intelligence
19 November 2009
Swarm Intelligence

Group Communications
Antenna Design
Cognitive radio MAC
Routing
Channel Modeling

Service Placement
Information Management
MIMO
LTE
DTN

Prof. Dr.-Ing. habil. Andreas Mitschele-Thiel
Integrated HW/SW Systems Group
Swarm Intelligence
19 November 2009
Overview

- Social insects
- Swarm Intelligence
  - Definition
  - Principles
  - Stigmergy
- Ant Colony Optimization
- Conclusion
- References
Social insects: Examples

Leafcutter ants (*Atta*)

Weaver ant (*Oecophylla*)

Wasp (*Polybia occidentalis*)
A social insect colony is...

- **Flexible**: The colony can respond to internal perturbations and external challenges.
- **Robust**: Tasks are completed even if some individuals fail.
- **Decentralized**: There is no central control in the colony.
- **Self-organized**: Paths to solutions are emergent rather than predefined.
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

- Ants
- Termites
- Bees
- Bird flocks
- Fish shoals
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),
  - ...

Prof. Dr.-Ing. habil. Andreas Mitschele-Thiel
Integrated HW/SW Systems Group

Swarm Intelligence
19 November 2009
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

\[ S_j : \text{Pillar construction state} \]
\[ R_j : \text{Response} \]
SI: Main application

- SI principles have been successfully applied in a variety of problem domains and applications:

  - **Ant colony optimization (ACO),**
    - Which focuses on discrete optimization problems

  - **Particle swarm optimization (PSO)**
    - Which focuses on nonlinear optimization problems with constraints
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)

Nest

Food

Ants start with equal probability
Foraging behavior of Ants
(Double bridge experiment)

The ant on shorter path has a shorter to-and-fro 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
Foraging behavior of Ants
(Double bridge experiment)

The next ant takes the shorter route
Foraging behavior of Ants
(Double bridge experiment)

After some time, the shorter path is almost exclusively used
From nature to computers

- The use of simple computational agents that work cooperatively
- In an iterative fashion, each ant moves from state $S_i$ to state $S_j$ guided by two main factors:

1. Heuristic information:
   - A measure of the heuristic preference for moving from state $S_i$ to state $S_j$
   - This information is known a priori to the algorithm run, and is not modified during

2. Artificial pheromone trail(s):
   - A measurement of the pheromone deposition from ants previous transitions from state $S_i$ to state $S_j$
   - This information is modified during the algorithm run by the artificial ants
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
What is Metaheuristic?

• “A metaheuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems”

• In other words: “a metaheuristic is a general-purpose algorithmic framework that can be applied to different optimization problems with relatively few modifications”

• Examples of metaheuristics:
  – Simulated annealing
  – Tabu search
  – Iterated local search
  – ...

...Dorigo, 1999
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
Ant System and the TSP

- The following steps is used to solve the TSP:
  - Pheromone trail
  - Memory
  - Awareness of environment
  - Probability function
Ant System and the TSP

1. Pheromone trail
   - Iteration is defined as the interval in \((t,t+1)\) where each of the \(N\) ants moves once
   - Epoch ➔ \(n\) 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 \( d_{ij} = \sqrt{\left( (x_i - x_j)^2 + (y_i - y_j)^2 \right)} \)

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

• Various improvements were made which gave rise to several other ant algorithms which collectively form the main ACO algorithms, such as:
  – Max–Min Ant System
  – Ant Colony System
  – …
Max–Min Ant System (MMAS)

• Developed by Stutzle and Hoos (1996)
• The differences between MMAS and AS are:
  – Only best ant updates pheromone
  – Pheromone trail values are restricted to an interval \([\tau_{\text{min}}, \tau_{\text{max}}]\)
  – Trails are initialized to their maximum value \(\tau_{\text{max}}\)
  – The modified pheromone update is as follows:
    \[
    \tau_{ij} \leftarrow \left[ (1 - \rho) \cdot \tau_{ij} + \Delta \tau_{ij}^{\text{best}} \right]_{\tau_{\text{min}}}^{\tau_{\text{max}}}
    \]
    where
    \[
    \Delta \tau_{ij}^{\text{best}} = \begin{cases} 
    \frac{1}{L_{\text{best}}} & \text{if } (i, j) \text{ belongs to the best tours} \\
    0 & \text{otherwise}
    \end{cases}
    \]
Ant Colony System (ACS)

• Developed by Dorigo & Gambardella (1997)
• The differences between ACS and AS are:
  – Pheromone update is done by all ants after each construction step only to last edge traversed.
  – Two pheromone update functions:
    • Local pheromone update: \( \tau_{ij} = (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0 \)
      where \( \varphi \in (0,1] \) is the pheromone decay function and \( \tau_0 \) is the initial value of the pheromone.
    • Offline pheromone update:
      \[
      \tau_{ij} = \begin{cases} 
      (1 - \rho) \cdot \tau_{ij} + \Delta \tau_{ij} & \text{if ant (i, j) belongs to best tour,} \\
      0 & \text{otherwise,}
      \end{cases}
      \]

Solving a Problem by ACO

Steps:

1. Represent the problem in the form a weighted graph, on which ants can build solutions
2. Define the meaning of the pheromone trails
3. Define the heuristic preference for the ant while constructing a solution
4. Choose a specific ACO algorithm and apply to problem being solved
5. Tune the parameters of the ACO algorithm
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

• SI:
  – 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
• ACO is a recently proposed metaheuristic approach for solving hard combinatorial optimization problems.
• Artificial ants implement a randomized construction heuristic which makes probabilistic decisions.
• The accumulated search experience is taken into account by the adaptation of the pheromone trail.
• ACO Shows great performance with the “ill-structured” problems like network routing.
• In ACO Local search is extremely important to obtain good results.
References

• Marco Dorigo “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem ” TR/IRIDIA/1996-5 Université Libre de Bruxelles, Belgium Swarm Intelligence: From Natural to Artificial System
Recommended Books

Swarm Intelligence
From Natural to Artificial Systems
Eric Bonabeau
Marco Dorigo
Guy Theraulaz

Swarm Intelligence
James Kennedy
Russell C. Eberhart

Computational Intelligence
An Introduction
Andries P. Engelbrecht

Prof. Dr.-Ing. habil. Andreas Mitschele-Thiel
Integrated HW/SW Systems Group
Swarm Intelligence
19 November 2009
Thanks...