Self-Organization in Autonomous Sensor/Actuator Networks

[SelfOrg]

Dr.-Ing. Falko Dressler

Computer Networks and Communication Systems
Department of Computer Sciences
University of Erlangen-Nürnberg

http://www7.informatik.uni-erlangen.de/~dressler/
dressler@informatik.uni-erlangen.de
Overview

- **Self-Organization**
  Basic methodologies of self-organization; comparison of central and hierarchical control, distributed systems, and autonomous behavior; examples of self-organization

- **Mobile Sensor/Actuator Networks**
  Ad hoc routing; reliable communication and congestion control; sensor assistance for mobile robots; applications

- **Coordination of Autonomous Systems**
  Coordination and synchronization; communication aspects; clustering

- **Bio-inspired Mechanisms**
  Swarm intelligence; artificial immune system; intra/inter cellular information exchange
Task and Resource Allocation

- (Multi-)Agent Systems
- Center-based algorithms
- Mediation algorithms
Task and Resource Allocation

- Problem description
  - Identify an appropriate system in a group of (autonomous) systems that
    - Has the required resources
    - These resources are available
    - The system is available to perform the requested task

- Types of resources
  - CPU capacity
  - Memory / storage
  - Energy
  - Time
  - Optimal position

- Allocation process
  - Identification of available nodes that show the required properties
    - election / negotiation / optimization
Dynamic Resource-Bounded Negotiation

- Negotiation protocols
  - Tasks can interact arbitrarily
  - Agents must negotiate the assignment of resources to tasks in dynamically changing environments
    - term negotiation to refer to any distributed process through which agents can agree on an efficient apportionment of tasks among themselves

- Assumptions commonly made in negotiation literature
  - Context in which a negotiation is made is irrelevant to the negotiation, thus task costs can be assumed to be additive
    - we allow for the possibility of positive and negative task interactions
  - The environment remains static during a negotiation
    - we allow for the possibility that important changes can occur during a negotiation that affect the result of the negotiation
  - All changes can be anticipated during negotiation
    - in any realistic domain, the world may change in unexpected ways
Examples

- Sensor challenge problem
  - If a deactivated emitter is activated, the beam is unstable and will not give reliable measurements for 2 seconds
  - If one task is immediately followed by another in the same sector, the beam will not require the 2 second warmup → this corresponds to **positive task interaction**

```plaintext
Arrival of task 1, Negotiation to S1
Arrival of task 2, negotiation to S1
0 2
Sensor S1
Sensor S2
```

- Multiple detectors
  - Consider that only one of three detectors on a sensor can be scanned at a given time and each scan takes between 0.6-1.8 seconds
  - Two sequential tasks that are less than 0.6 seconds apart and occur in separate sectors will **interact negatively**

```plaintext
Arrival of task 1, Negotiation to S1
Arrival of task 2, negotiation to S2
0 0.6
Sensor S1
Sensor S2
```
Center-Based Algorithms

- Working principle
  - Center agent collects bids on proposed allocations
  - Each bid is meant to compactly encapsulate important local information
  - Decision is taken by the center-agent

- Remark: *there is a difference to central coordinator!*
  - Central coordinator has up-to-date information regarding the local states of the agents; used to compute optimal allocations
  - Amount of information (local state of the agents) may be large
    → infeasible in cases of communication limits and system faults
Center-Based Algorithms

- Formal definition
  - Task allocation system: $M = <A, T, u, P>$
  - $A = \{a_1, \ldots, a_n\}$ is a set of $n$ agents with some agent designated as the mediator
  - $T = \{t_1, \ldots, t_m\}$ is a set of $m$ tasks
  - $u: A \times 2^T \rightarrow \mathbb{R} \cup \{\infty\}$ is a value function that returns the value which an agent associates with a particular subset of tasks
  - $P$ is an assignment (or partition) of size $n$ on the sets of tasks $T$ such that $P = <P_1, \ldots, P_n>$, where $P_j$ contains the set of items assigned to agent $a_j$
  - We refer to $P$ as a proposal; for example $P_5 = <a_1, a_5, a_3>$ corresponds to the allocation in which task $t_1$ is assigned to agent $a_1$, $t_2$ to $a_5$, and $t_3$ to $a_3$

  - The objective function $f$ determines the desirability of an assignment based on the values that each agent ascribes to the items it is assigned

$$f(p, A) = \sum_{a \in A} u(a, p) \quad p \in P$$
Center-Based Algorithms

- Formal definition (cont’d.)
  - The *negotiation problem* is that of choosing an element $p^*$ of $P$ that maximizes the objective function

$$p^* = \arg \max_{p \in P} f(p, A)$$

- The proposal chosen is called the *outcome* of the negotiation

- Both, mediation and combinatorial auctions are examples of algorithms that can be used to solve the assignment problem
  - class of center-based assignments (CBA)
Auctions

- Sequential auctions? (serialized item allocation)
  - Simple building rules
  - Provide no context (list of other tasks to which an agent will be assigned in later auctions)
  - Assumptions must be made about the outcomes of other, related auctions

- Combinatorial auctions? (for exploring allocations of items that interact
  - agents have the freedom to choose particular bunches of items)
  - Allow an agent to pick certain bundles of tasks which might interact in a favorable way
  - Introduce a bid generation problem

→ re-allocation might help to solve these issues
Mediation Algorithm

- **Basic idea**
  - An agent is selected to act as mediator
  - It implements a *hill-climbing search* in the proposal space
  - Use of a communication channel
    (costly in terms of time, etc. but assumed to be lossless)

- **Mediation algorithm**
  - Inputs: P, A, update procedure such as AIM (allocation improvement mediation)
  - Supports group decisions

  - The algorithm is anytime: it can be halted at any time and will return the best proposal found so far
  - Therefore, the mediation is applicable even if the agents do not know in advance how much time they will have to negotiate
Mediation Algorithm

function MEDIATION returns an outcome
inputs: \( P \), \( A \), UpdateProcedure

let \( b \leftarrow 0 \), \( b_{\text{val}} \leftarrow \text{VALUE}(0) \)
loop
    \( c \leftarrow \text{next value generated by UpdateProcedure} \)
    broadcast \( c \) to \( A \)
    for each \( a_i \) in \( A \)
        receive \( \text{msg}_i \) from \( a_i \)
        \( c_{\text{val}} \leftarrow \text{VALUE}(\text{msg}_1, \text{msg}_2, \ldots, \text{msg}_n) \)
    if \((c_{\text{val}} > b_{\text{val}})\) then
        \( b \leftarrow c, b_{\text{val}} \leftarrow c_{\text{val}} \)
until (stop signal)
return \( b \)

1. Mediator initializes \( b \) (representing the best proposal found so far) along with an initial value
2. An update procedure generates another proposal \( c \) (current proposal)
3. This proposal is broadcast to the group
4. Each agent responds with a message \( \text{msg}_i \) based on the proposal \( c \)
5. Messages are combined to form a value
6. If the value is preferred to the current \( b_{\text{val}} \), \( b \) is updated with the current proposal
Allocation Improvement

- Update procedure for mediation that supports task allocation domains

  let p ← a random element of P - {0}; return p
  for i = 1 … |T|
    for t ← every set of tasks of size I
      for a ← every possible assignment of agents in A to tasks in t
        q ← substitute a in p; return q
        if qval > pval in mediation then p ← q

- The first proposal p is chosen randomly from P
  - The proposal provides a context, from which subsequent proposals are generated, e.g. it might return <{t2},{t0,t1}>, i.e. agent 0 is assigned task 2 and agent 1 to tasks 0 and 1
  - This context is common to all agents and ensures that each task is assigned to an agent

- Subsequent iterations
  - the procedure returns proposals that result from making substitutions in p for i-tuples of tasks where i goes from 1 to |T|
  - p is always maintained to correspond to the best proposal in mediation
Experimental Analysis

- Allocation Improvement Mediation
- Random Mediation (returns a random element of \( P \) at each iteration)
- Full Search (simply returns successive elements of \( P \))

4-agent sensor domain

20-agent sensor domain

[SelfOrg], SS 2006
Task Allocation: Where to go?

- So far, only sets with static resources have been investigated into, what about the possibility to let tasks and resources dynamically appear and disappear?

- First solution (usually found in the literature): the ongoing negotiation is interrupted / a re-allocation is initiated.

- More practicable (and more sophisticated): dynamic mediation
  - a mixture of central coordination and mediation
  - The bids are enriched to include all relevant local state information
    → a negotiation space is available at the mediator (set of resources and tasks)
  - This negotiation space might change because of
    - A negotiation event (the mediator considers a new resource)
    - A domain event (a new task appears)