The use of Organizational Self Design to coordinate multiagent systems

Sachin Kamboj

Department of Computer and Information Sciences
University of Delaware
Newark, DE 19716

Ph.D. Proposal Presentation
Outline

1. Introduction
   - Motivation
   - Problem Representation

2. Approach
   - Agent roles and relationships
   - Organization Formation and Adaptation

3. Progress
   - Work done till date
   - Work currently in progress
   - Future work planned
Outline

1 Introduction
   - Motivation
   - Problem Representation

2 Approach
   - Agent roles and relationships
   - Organization Formation and Adaptation

3 Progress
   - Work done till date
   - Work currently in progress
   - Future work planned
Introduction
Research Question

How many agents does it take to screw in a light bulb?

Answer: 42
- 3 to negotiate over which type of light-bulb to buy...
- 5 to coordinate the buying action...
How many agents does it take to screw in a light bulb?

Answer: 42

- 3 to negotiate over which type of light-bulb to buy...
- 5 to coordinate the buying action...
How many agents does it take to screw in a light bulb?

Answer: 42

- 3 to negotiate over which type of light-bulb to buy...
- 5 to coordinate the buying action...

::

::
Basic Model

Problem Instance
Instance # 1
Deadline: 40
Task:
Method 5
Method 4
Method 3
Method 2
Task D
Method 1
Task B Task C
Task A
Method 6
TIME
MULTIAGENT SYSTEM
SOLUTIONS
Multiagent system has to be designed at two levels:

- **Micro-architectural level**
  - Involves the design of the individual agents
- **Macro-architectural level**
  - Focus of my research
Motivation

Multiagent system has to be designed at two levels:

- **Micro-architectural level**
  - Involves the design of the individual agents

- **Macro-architectural level**
  - Focus of my research
Motivation

Macro-architectural level

- How many agents do we need?
- How do we allocate subtasks and resources to the agents?
- How do we coordinate inter-agent activities?

Design an organization for the agents!!!
Motivation

Macro-architectural level

- How many agents do we need?
- How do we allocate subtasks and resources to the agents?
- How do we coordinate inter-agent activities?

Design an organization for the agents!!!
Macro-architectural level

- How many agents do we need?
- How do we allocate subtasks and resources to the agents?
- How do we coordinate inter-agent activities?

Design an organization for the agents!!!
Motivation

Macro-architectural level

- How many agents do we need?
- How do we allocate subtasks and resources to the agents?
- How do we coordinate inter-agent activities?

Design an organization for the agents!!!
What is an organization

- No universally accepted definition
- Organizations emerge whenever agents work together in a shared environment. Organizations reflect the structure of the interactions of the participating agents and are engaged in tasks and goal oriented behavior.  
  
  (Singh.05)

- Organizations are a consciously coordinated social entity, with a relatively identifiable boundary, that functions on a relatively continuous basis to achieve a common goal or a set of goals.  
  
  (Robbins.89)
What is an organization

No universally accepted definition

Organizations emerge whenever agents work together in a shared environment. Organizations reflect the structure of the interactions of the participating agents and are engaged in tasks and goal oriented behavior. (Singh.05)

Organizations are a consciously coordinated social entity, with a relatively identifiable boundary, that functions on a relatively continuous basis to achieve a common goal or a set of goals. (Robbins.89)
What is an organization

- No universally accepted definition
- Organizations emerge whenever agents work together in a shared environment. Organizations reflect the structure of the interactions of the participating agents and are engaged in tasks and goal oriented behavior. (Singh.05)
- Organizations are a consciously coordinated social entity, with a relatively identifiable boundary, that functions on a relatively continuous basis to achieve a common goal or a set of goals. (Robbins.89)
What is an organization

- A long-term commitment made by the agents to a particular way of handling the cooperative tasks. (So and Durfee.93)
- Something that binds agents together to achieve effective coordination towards some common goal. (Barber.01)
What is an organization

A long-term commitment made by the agents to a particular way of handling the cooperative tasks. (So and Durfee.93)

Something that binds agents together to achieve effective coordination towards some common goal. (Barber.01)
What is an organization

Organizations (adapted from Carley and Gasser.99):

- are composed of cooperative agents
- goal directed
- allow effective coordination
- are reasonably long term in duration:
- able to affect and be affected by the environment
- are characterized by an organizational structure
What is an organization

Organizations (adapted from Carley and Gasser.99):

- are composed of cooperative agents
- goal directed
  - allow effective coordination
- are reasonably long term in duration:
  - able to affect and be affected by the environment
- are characterized by an organizational structure
What is an organization

Organizations (adapted from Carley and Gasser.99):

- are composed of cooperative agents
- goal directed
- allow effective coordination
- are reasonably long term in duration:
- able to affect and be affected by the environment
- are characterized by an organizational structure
What is an organization

Organizations (adapted from Carley and Gasser.99):

- are composed of cooperative agents
- goal directed
- allow effective coordination
- are reasonably long term in duration:
  - able to affect and be affected by the environment
  - are characterized by an organizational structure
What is an organization

Organizations (adapted from Carley and Gasser.99):

- are composed of cooperative agents
- goal directed
- allow effective coordination
- are reasonably long term in duration:
  - able to affect and be affected by the environment
- are characterized by an organizational structure
Organizations (adapted from Carley and Gasser.99):
- are composed of cooperative agents
- goal directed
- allow effective coordination
- are reasonably long term in duration:
- able to affect and be affected by the environment
- are characterized by an organizational structure
Constituents of an organization

According to Coutinho et. al., organizations consist of:

1. structural dimension
2. functional dimension
3. dialogical dimension
4. deontic dimension
According to Coutinho et. al., organizations consist of:

1. structural dimension
2. functional dimension
3. dialogical dimension
4. deontic dimension
According to Coutinho et. al., organizations consist of:

1. structural dimension
2. functional dimension
3. dialogical dimension
4. deontic dimension
According to Coutinho et. al., organizations consist of :

1. structural dimension
2. functional dimension
3. dialogical dimension
4. deontic dimension
Constituents of an organization

For my research, organizations consist of:

- an organizational structure:
  - roles
  - relationships

- coordination mechanisms
Constituents of an organization

For my research, organizations consist of:

- an organizational structure:
  - roles
  - relationships
- coordination mechanisms
Constituents of an organization

For my research, organizations consist of:

- an organizational structure:
  - roles
  - relationships

- coordination mechanisms
Constituents of an organization

For my research, organizations consist of:

- an organizational structure:
  - roles
  - relationships
- coordination mechanisms
Contingency Theory (Lawrence and Lorsch)

- There is no best way to organize
- All ways of organizing are not equally effective
Designing an organization

Optimal organizational structure depends on:

- Problem being solved
- Environmental conditions under which the problem is being solved
Approaches to organizational design

- Static organizational design:
  - AGR (Ferber et. al.)
  - ODML (Horling and Lesser)
  - OMNI (Dignum et. al.)
  - MOISE+ (Hubneret. al.)
  - ISLANDER (Esteva et al.)
Designing an organization

Problems with static organizational design:

- Environmental conditions may not be known *a priori*
- Environment might be dynamic
  - Problems arrive at varying rates
  - Problems might have different deadlines
  - Available resources might change over time

Solution: Design an organization at run-time
Problems with static organizational design:

- Environmental conditions may not be known \textit{a priori}
- Environment might be dynamic
  - Problems arrive at varying rates
  - Problems might have different deadlines
  - Available resources might change over time

Solution: Design an organization at run-time
Designing an organization

Problems with static organizational design:
- Environmental conditions may not be known *a priori*
- Environment might be dynamic
  - Problems arrive at varying rates
  - Problems might have different deadlines
  - Available resources might change over time

Solution: Design an organization at run-time
Designing an organization

Problems with static organizational design:

- Environmental conditions may not be known \textit{a priori}
- Environment might be dynamic
  - Problems arrive at varying rates
  - Problems might have different deadlines
  - Available resources might change over time

Solution: Design an organization at run-time
Designing an organization

Problems with static organizational design:

- Environmental conditions may not be known \textit{a priori}
- Environment might be dynamic
  - Problems arrive at varying rates
  - Problems might have different deadlines
  - Available resources might change over time

Solution: Design an organization at run-time
Problems with static organizational design:

- Environmental conditions may not be known \textit{a priori}
- Environment might be dynamic
  - Problems arrive at varying rates
  - Problems might have different deadlines
  - Available resources might change over time

Solution: Design an organization at run-time
Designing an organization

Problems with static organizational design:

- Environmental conditions may not be known *a priori*
- Environment might be dynamic
  - Problems arrive at varying rates
  - Problems might have different deadlines
  - Available resources might change over time

Solution: Design an organization at run-time
Designing an organization

Cool, we’ll just use the Contract Net Protocol (Smith and Davis.78)
Designing an organization

Contract Net Protocol

![Contract Net Protocol Diagram]

- Task A
- Task B
- Task C

Method 1
Method 2
Method 3
Method 4

Agent 1
Designing an organization

Contract Net Protocol

Agent 1

Request for bids (Task B)

Agent 2

Agent 3

Agent 4

Introduction
Motivation
Problem
Representation

Approach
Agent roles and relationships
Organization
Formation and Adaptation

Progress
Work done till date
Work currently in progress
Future work planned
Designing an organization

Contract Net Protocol

Agent 1

Bid (Task B)

Agent 2

Agent 3

Agent 4
Designing an organization

Contract Net Protocol

Agent 1

Award (Task B)

Agent 2

Agent 3

Agent 4

Introduction
Motivation
Problem
Representation
Approach
Agent roles and relationships
Organization
Formation and Adaptation
Progress
Work done till date
Work currently in progress
Future work planned
Contract Net Protocol

Request for bids (Method 1)

Agent 1

Agent 2

Agent 3

Agent 4
Designing an organization

Problems

Problem 1: Build Car 1
Problem 2: Build Car 2
Problem 3: Build Car 3
Problem 4: Build Car 4

Instance of
Instance of
Instance of
Instance of

Build a Car

Build the parts
Assemble the Car
Build the engine
Frame the body
Build the transmission
Designing an organization

Contract Net Protocol has some serious flaws:
- designs a new organizational structure for *every problem instance*...
  - all instances are not unique
  - environment for each instance is not unique
- organizational structure is implicit
Designing an organization

Contract Net Protocol has some serious flaws:

- designs a new organizational structure for every *problem instance*...
  - all instances are not unique
  - environment for each instance is not unique
- organizational structure is implicit
Contract Net Protocol has some serious flaws:

- designs a new organizational structure for every *problem instance*...
  - all instances are not unique
  - environment for each instance is not unique

- organizational structure is implicit
Designing an organization

Contract Net Protocol has some serious flaws:

- designs a new organizational structure for every problem instance...
  - all instances are not unique
  - environment for each instance is not unique
- organizational structure is implicit
Designing an organization

Our Answer:

Organizational Self-Design (Gasser, Ishida, et. al.):

- involves designing an organization at run-time
- agents are responsible for designing their own organizational structures
- organizational structure is explicit
- organizational structure persists across problem instances
Our Answer: 
Organizational Self-Design (Gasser, Ishida, et. al.):
- involves designing an organization at run-time
- agents are responsible for designing their own organizational structures
  - organizational structure is explicit
  - organizational structure persists across problem instances
Our Answer:

**Organizational Self-Design** (Gasser, Ishida, et. al.):

- involves designing an organization at run-time
- agents are responsible for designing their own organizational structures
- organizational structure is **explicit**
- organizational structure persists across problem instances
Our Answer:

Organizational Self-Design (Gasser, Ishida, et. al.):
- involves designing an organization at run-time
- agents are responsible for designing their own organizational structures
- organizational structure is explicit
- organizational structure persists across problem instances
Designing an organization

Organizational self-design

Problem Instances → Multiagent organization → SOLUTIONS

Compute organization performance

If performance < desired

Generate new organization
How can we generate a new organization?

- Change the number of agents
  - Shehory et. al. 98, Gasser and Ishida. 91

- Change the role allocation
  - Gasser and Ishida. 91, Goldman and Rosenchein. 97, Dignum et. al. 04

- Change the relationships
  - Barber. et. al. 01

- Change the coordination mechanism
  - Barber. et. al. 01
How can we generate a new organization?

- Change the number of agents
  - Shehory et. al.98, Gasser and Ishida. 91
- Change the role allocation
  - Gasser and Ishida. 91, Goldman and Rosenchein.97, Dignum et. al.04
- Change the relationships
  - Barber. et. al.01
- Change the coordination mechanism
  - Barber. et. al.01
How can we generate a new organization?

- Change the number of agents
  - Shehory et. al.98, Gasser and Ishida. 91

- Change the role allocation
  - Gasser and Ishida. 91, Goldman and Rosenchein.97, Dignum et. al.04

- Change the relationships
  - Barber. et. al.01

- Change the coordination mechanism
  - Barber. et. al.01
How can we generate a new organization?

- Change the number of agents
  - Shehory et. al. 98, Gasser and Ishida. 91
- Change the role allocation
  - Gasser and Ishida. 91, Goldman and Rosenchein. 97, Dignum et. al. 04
- Change the relationships
  - Barber. et. al. 01
- Change the coordination mechanism
  - Barber. et. al. 01
How can we generate a new organization?

- Change the number of agents
  - Shehory et. al. 98, Gasser and Ishida. 91

- Change the role allocation
  - Gasser and Ishida. 91, Goldman and Rosenchein.97, Dignum et. al.04

- Change the relationships
  - Barber. et. al.01

- Change the coordination mechanism
  - Barber. et. al.01
Categorization of problem domains by Rosenschein and Zlotkin
Problem Domains

Task oriented domains

Book 1
Book 2
Book 3
Book 4
Book 5
Book 6
Problem Domains

State oriented domains

- Book 1
- Book 2
- Book 6
- Book 1
- Book 2
- Book 6
- Book 1
- Book 5
- Book 6
- Book 1
- Book 5
- Book 6
- Book 1
- Book 3
- Book 6
- Book 3
- Book 4
- Book 6
- Book 3
- Book 4
- Book 6
My research focuses on worth-oriented domains!!!
Problems are represented using TÆMS

Method 1
Outcome: 1
Probability: 1.0
Quality: 6
Cost: 7
Duration: 32

Method 4
Outcome: 1
Probability: 0.5
Quality: 3
Cost: 4
Duration: 2
Outcome: 2
Probability: 0.5
Quality: 12
Cost: 10
Duration: 3

Method 2
Outcome: 1
Probability: 1.0
Quality: 10
Cost: 5
Duration: 17

Method 5
Outcome: 1
Probability: 0.7
Quality: 10
Cost: 5
Duration: 20
Outcome: 2
Probability: 0.3
Quality: 3
Cost: 7
Duration: 30

Method 3
Outcome: 1
Probability: 1.0
Quality: 12
Cost: 7
Duration: 12

Method 6
Outcome: 1
Probability: 1.0
Quality: 12
Cost: 7
Duration: 12
Problems are represented using TÆMS
Problem Representation

Task A

Task B

Task C

Task D

Method 1

Method 2

Method 3

Method 4

Method 5

Method 6
Problem Representation

TÆMS is quantitative

- Task A
  - Task B
    - Method 1
      - Outcome: 1
      - Probability: 1.0
      - Quality: 6
      - Cost: 7
      - Duration: 32
  - Task D
    - Method 5
      - Outcome: 1
      - Probability: 0.7
      - Quality: 10
      - Cost: 5
      - Duration: 20
      - Outcome: 2
      - Probability: 0.3
      - Quality: 3
      - Cost: 7
      - Duration: 30
    - Method 6
      - Outcome: 1
      - Probability: 1.0
      - Quality: 12
      - Cost: 10
      - Duration: 3
    - Method 4
      - Outcome: 1
      - Probability: 0.5
      - Quality: 3
      - Cost: 4
      - Duration: 2
      - Outcome: 2
      - Probability: 0.5
      - Quality: 12
      - Cost: 10
      - Duration: 3
  - Task C
    - Method 2
      - Outcome: 1
      - Probability: 1.0
      - Quality: 10
      - Cost: 5
      - Duration: 17
    - Method 3
      - Outcome: 1
      - Probability: 0.7
      - Quality: 10
      - Cost: 5
      - Duration: 20
      - Outcome: 2
      - Probability: 0.3
      - Quality: 3
      - Cost: 7
      - Duration: 30
Problem Representation

TÆMS is quantitative

Method 1
Outcome: 1
Probability: 1.0
Quality: 6
Cost: 7
Duration: 32

Method 2
Outcome: 1
Probability: 1.0
Quality: 10
Cost: 5
Duration: 17

Method 3
Outcome: 1
Probability: 1.0
Quality: 12
Cost: 7
Duration: 12

Method 4
Outcome: 1
Probability: 0.5
Quality: 3
Cost: 4
Duration: 2
Outcome: 2
Probability: 0.5
Quality: 12
Cost: 10
Duration: 3

Method 5
Outcome: 1
Probability: 0.7
Quality: 10
Cost: 5
Duration: 20
Outcome: 2
Probability: 0.3
Quality: 3
Cost: 7
Duration: 30

Method 6
Outcome: 1
Probability: 1.0
Quality: 12
Cost: 7
Duration: 7

Method 7
Outcome: 1
Probability: 0.5
Quality: 3
Cost: 4
Duration: 2
Outcome: 2
Probability: 0.5
Quality: 12
Cost: 10
Duration: 3

Method 8
Outcome: 1
Probability: 0.7
Quality: 10
Cost: 5
Duration: 20
Outcome: 2
Probability: 0.3
Quality: 3
Cost: 7
Duration: 30

Method 9
Outcome: 1
Probability: 1.0
Quality: 12
Cost: 7
Duration: 7

Method 10
Outcome: 1
Probability: 0.5
Quality: 3
Cost: 4
Duration: 2
Outcome: 2
Probability: 0.5
Quality: 12
Cost: 10
Duration: 3

Method 11
Outcome: 1
Probability: 0.7
Quality: 10
Cost: 5
Duration: 20
Outcome: 2
Probability: 0.3
Quality: 3
Cost: 7
Duration: 30

Method 12
Outcome: 1
Probability: 1.0
Quality: 12
Cost: 7
Duration: 7
Problem Representation

- **Introduction**
- **Motivation**
- **Problem Representation**
- **Approach**
  - Agent roles and relationships
  - Organization
  - Formation and Adaptation
- **Progress**
  - Work done till date
  - Work currently in progress
  - Future work planned

### Problem Representation

**Task A**
- Methods:
  - **Method 1**
    - Outcome: 1
    - Probability: 1.0
    - Quality: 6
    - Cost: 7
    - Duration: 32
  - **Method 2**
    - Outcome: 1
    - Probability: 1.0
    - Quality: 10
    - Cost: 5
    - Duration: 17
  - **Method 3**
    - Outcome: 1
    - Probability: 0.7
    - Quality: 10
    - Cost: 5
    - Duration: 20
  - **Method 4**
    - Outcome: 1
    - Probability: 0.5
    - Quality: 3
    - Cost: 4
    - Duration: 2
    - Outcome: 2
    - Probability: 0.5
    - Quality: 12
    - Cost: 10
    - Duration: 3
  - **Method 5**
    - Outcome: 1
    - Probability: 0.7
    - Quality: 10
    - Cost: 5
    - Duration: 20
  - **Method 6**
    - Outcome: 1
    - Probability: 1.0
    - Quality: 12
    - Cost: 7
    - Duration: 7

### Approach

- **Agent roles and relationships**
- **Organization**
- **Formation and Adaptation**

---

**Facilitates**

**Enables**

**MAX**

**SUM**

**SEQ**

**MIN**
Outline

1. Introduction
   - Motivation
   - Problem Representation

2. Approach
   - Agent roles and relationships
   - Organization Formation and Adaptation

3. Progress
   - Work done till date
   - Work currently in progress
   - Future work planned
Organizational design is directly contingent on:

1. The task structure
2. The environmental conditions under which the problems need to be solved
Agent roles and relationships

- Organizational structure is primarily composed of
  - Roles:
    - Parts played by the agents enacting the roles in the solution to the problem
  - Relationships:
    - Coordination relationships that exist between the subparts of a problem

Definition

A role is a TÆMS subtree rooted at a particular node
Agent roles and relationships

Organizational structure is primarily composed of

Roles:
- Parts played by the agents enacting the roles in the solution to the problem

Relationships:
- Coordination relationships that exist between the subparts of a problem

Definition
A role is a TÆMS subtree rooted at a particular node
Agent roles and relationships

- Organizational structure is primarily composed of:
  - Roles:
    - Parts played by the agents enacting the roles in the solution to the problem
  - Relationships:
    - Coordination relationships that exist between the subparts of a problem

Definition

A role is a TÆMS subtree rooted at a particular node
Agent roles and relationships

Organizational structure is primarily composed of:

- **Roles:**
  - Parts played by the agents enacting the roles in the solution to the problem

- **Relationships:**
  - Coordination relationships that exist between the subparts of a problem

**Definition**

A role is a TÆMS subtree rooted at a particular node
Agent roles and relationships

- Task A
  - MAX
  - Enables
    - Task B
      - SUM
      - Method 1
        - Method 4
        - Method 5
        - Method 6
      - Facilitates
        - MIN
        - Method 2
        - Method 3
  - Enables
    - Task C
      - SEQ_SUM
      - Method 2
      - Method 3
Agent roles and relationships

Task A

Task B
- Method 1
- Task D
  - Method 4
  - Method 5
  - Method 6

Task C
- Method 2
- Method 3

Role 1
- Enables MAX
- SUM
- Facilitates MIN

Role 2
- Enables SEQ_SUM

Method 1
Method 2
Method 3
Method 4
Method 5
Method 6
Agent roles and relationships

Task A

Task B

Task C

Task D

Role 1

Role 2

Role 3

Role 4

Method 2

Method 3

Method 4

Method 5

Method 6

MAX

SUM

SEQ_SUM
Outline

1. Introduction
   - Motivation
   - Problem Representation

2. Approach
   - Agent roles and relationships
   - Organization Formation and Adaptation

3. Progress
   - Work done till date
   - Work currently in progress
   - Future work planned
Organization Formation and Adaptation

Start off with an initial organization consisting of a single agent, solely responsible for all activities.

Each agent in the organization checks to see if:
- It is overloaded:
  - It spawns off a new agent to handle part of its load
- It is free (underloaded)
  - It combines with another agent to save resources
Start off with an initial organization consisting of a single agent, solely responsible for all activities.

Each agent in the organization checks to see if:

- It is overloaded:
  - It spawns off a new agent to handle part of its load.
- It is free (underloaded):
  - It combines with another agent to save resources.
Organization Formation and Adaptation

Start off with an initial organization consisting of a single agent, solely responsible for all activities.

Each agent in the organization checks to see if:

- It is overloaded:
  - It spawns off a new agent to handle part of its load
  - It is free (underloaded)
- It combines with another agent to save resources
Organization Formation and Adaptation

- Start off with an initial organization consisting of a single agent, solely responsible for all activities
- Each agent in the organization checks to see if:
  - It is overloaded:
    - It spawns off a new agent to handle part of its load
  - It is free (underloaded)
    - It combines with another agent to save resources
Organization Formation and Adaptation

- Start off with an initial organization consisting of a single agent, solely responsible for all activities.
- Each agent in the organization checks to see if:
  - It is overloaded:
    - It spawns off a new agent to handle part of its load.
  - It is free (underloaded):
    - It combines with another agent to save resources.
Organization Formation and Adaptation

- Start off with an initial organization consisting of a single agent, solely responsible for all activities.
- Each agent in the organization checks to see if:
  - It is overloaded:
    - It spawns off a new agent to handle part of its load.
  - It is free (underloaded):
    - It combines with another agent to save resources.
Organization Formation and Adaptation
Agent Spawning

Agent Spawning

Facilitates

Method 1
Method 2
Method 3
Facilitates

Method 4
Method 5
Method 6
Organization Formation and Adaptation
Agent Spawning

Agent Spawning

Method 1
Facilitates
AGENT 1
Organization 1
Cost (Task B) = 10

Task A
MAX

Task B
SUM

Task C
SEQ_SUM

Task D
MIN

Method 2
Method 3

Method 4
Method 5
Method 6
Organization Formation and Adaptation
Agent Spawning
Introduction

Motivation

Problem

Representation

Approach

Agent roles and relationships

Organization formation and adaptation

Organization formation and adaptation

Agent spawning

Task A

Task B

Task C

Task D

Method 1

Method 2

Method 3

Method 4

Method 5

Method 6

Cost (Method 1) = 7
Organization Formation and Adaptation

Agent Spawning

- **INTRODUCTION**
  - Motivation
  - Problem
  - Representation

- **APPROACH**
  - Agent roles and relationships
  - Organization Formation and Adaptation

- **PROGRESS**
  - Work done till date
  - Work currently in progress
  - Future work planned

---

**Organization Formation and Adaptation**

**Agent Spawning**

**MIN**

**T**ask

**A**

**T**ask B

**Task C**

**Task D**

**Method 1**

**Method 2**

**Method 3**

**Method 4**

**Method 5**

**Method 6**

**SE**Q_**SUM**

**MAX**

**SUM**

**MIN**

**Enables**

**Facilitates**

Cost (Task D) = 8
Organization Formation and Adaptation
Agent Spawning

AGENT 1

Task A

Task B

Task C

Task D

Task A

Task B

Task C

Task D

Method 1

Method 2

Method 3

Method 4

Method 5

Method 6

Lowest Cost = Cost(Task C)
Organization Formation and Adaptation
Agent Spawning

Introduction
Motivation
Problem
Representation
Approach
Agent roles and relationships
Organization Formation and Adaptation
Progress
Work done till date
Work currently in progress
Future work planned

MIN
Task A
Task B
Task C
Task D
Method 1
Method 2
Method 3
Method 4
Method 5
Method 6

MAX
SUM
SEQ_SUM

AGENT 1
AGENT 2
Organization 2
Organization Formation and Adaptation
Agent Spawning

What is the cost function?

- We evaluated three cost functions (based on three heuristics):
  1. Allocating top-most roles first
  2. Minimizing total resource cost
  3. Balancing execution time
The three cost functions

**AGENT 1**

- **Task A**
  - **SUM**
  - **MIN**

- **Task B**
  - **MIN**
  - **Method 1**
    - Duration: 3
    - Resource A: 5
  - **Method 2**
    - Duration: 10
    - Resource A: 9

- **Task C**
  - **MIN**
  - **Method 3**
    - Duration: 2
    - Resource A: 7
  - **Method 4**
    - Duration: 4
    - Resource B: 5
**Topmost First**

- Number the roles while doing a BFS.
- $\text{Cost}(\text{Role}) = \text{Number assigned to Role}$

---

**Agent Spawning**

**Topmost First**

- Number the roles while doing a BFS.
- Cost(\text{Role}) = Number assigned to \text{Role}

---

**Method 1**

- Duration: 3
- Resource A: 5

**Method 2**

- Duration: 10
- Resource A: 9

**Method 3**

- Duration: 2
- Resource A: 7

**Method 4**

- Duration: 4
- Resource B: 5
**Organization Formation and Adaptation**

**Agent Spawning**

**Topmost First**

- Number the roles while doing a BFS.
- \( \text{Cost} (\text{Role}) = \text{Number assigned to Role} \)

![Diagram showing agent spawning with tasks and methods]

**AGENT 1**

- **Task A**
  - SUM
  - MIN

- **Task B**
  - MIN

- **Task C**
  - MIN

**Methods**

1. **Method 1**
   - Duration: 3
   - Resource A: 5

2. **Method 2**
   - Duration: 10
   - Resource A: 9

3. **Method 3**
   - Duration: 2
   - Resource A: 7

4. **Method 4**
   - Duration: 4
   - Resource B: 5
Topmost First

- Number the roles while doing a BFS.
- $\text{Cost}(Role) = \text{Number assigned to Role}$
Minimizing Resource Cost

- Cost(\textit{Role}) = \text{Resource Cost}(\textit{Role}) + \text{Resource Cost}(\textit{Role})

**Diagram:**

- **AGENT 1**
  - Task A
    - Task B
      - Method 1
        - Duration: 3
        - Resource A: 5
      - Method 2
        - Duration: 10
        - Resource A: 9
    - Method 3
      - Duration: 2
      - Resource A: 7
    - Method 4
      - Duration: 4
      - Resource B: 5
Minimizing Resource Cost

\[
\text{Cost}(\text{Role}) = \text{Resource Cost}(\text{Role}) + \text{Resource Cost}(\text{Role})
\]
Minimizing Resource Cost

Cost(Role) = Resource Cost(Role) + Resource Cost(Role)

-task A
-task B
-task C

Method 1
Duration: 3
Resource A: 5

Method 2
Duration: 10
Resource A: 9

Method 3
Duration: 2
Resource A: 7

Method 4
Duration: 4
Resource B: 5

AGENT 1

RC=9

Cost(Task B) = 9 + 12 = 21
Minimizing Resource Cost

\[ \text{Cost}(\text{Role}) = \text{Resource Cost}(\text{Role}) + \text{Resource Cost}(\text{Role}) \]

AGENT 1

Task A

Task B

Task C

Method 1

Duration: 3
Resource A: 5

Method 2

Duration: 10
Resource A: 9

Method 3

Duration: 2
Resource A: 7

Method 4

Duration: 4
Resource B: 5

Cost(Method 4) = 5 + 9 = 14
Minimizing Resource Cost

\[ \text{Cost}(Role) = \text{Resource Cost}(Role) + \text{Resource Cost}(Role) \]
Balancing Execution Time

Cost(Role) = | Expt Duration(Role) - Expt Duration(Role) |

Diagram:
- Task A
  - Task B
    - Method 1: Duration: 3, Resource A: 5
    - Method 2: Duration: 10, Resource A: 9
  - Task C
    - Method 3: Duration: 2, Resource A: 9
    - Method 4: Duration: 4, Resource B: 5

SUM, MIN, MIN
Balancing Execution Time

Cost(Role) = | Expt Duration(Role) - Expt Duration(Role) |

<table>
<thead>
<tr>
<th>Method</th>
<th>Duration</th>
<th>Resource A:</th>
<th>Resource B:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method 1</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Method 2</td>
<td>10</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Method 3</td>
<td>2</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Method 4</td>
<td>4</td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

AGENT 1

ED=13

ED=6
Balancing Execution Time

Cost(Role) = | Expt Duration(Role) - Expt Duration(Role) |

Cost(Task B) = 13 - 6 = 7
Balancing Execution Time

\[ \text{Cost}(\text{Role}) = | \text{Expt Duration}(\text{Role}) - \text{Expt Duration}(\text{Role}) | \]

**Task A**
- **Agent 1**
  - **Task B**
    - **Method 1**
      - Duration: 3
      - Resource A: 5
  - **Task C**
    - **Method 2**
      - Duration: 10
      - Resource A: 9
    - **Method 3**
      - Duration: 2
      - Resource A: 7
    - **Method 4**
      - Duration: 4
      - Resource B: 5
  - **SUM**
  - **MIN**

**Conclusion**
- The balancing execution time for the tasks is achieved through the methods and their respective durations and resources.
Balancing Execution Time

\[ \text{Cost}(Role) = | \text{Expt Duration}(Role) - \text{Expt Duration}(Role) | \]
Organization Formation and Adaptation
Agent Composition

AGENT 1

Task A

MAX

Task B

SUM

Task D

MIN

Method 1

Method 2

Method 3

Method 4

Method 5

Method 6

AGENT 2

Task C

SEQ_SUM

AGENT 3

AGENT 4

Organization i
Organization Formation and Adaptation
Agent Composition

Task A
Task B
Task C
Task D

Method 1
Method 2
Method 3
Method 4
Method 5
Method 6

AGENT 1
AGENT 2
AGENT 4

Facilitates
Enables

MAX
SUM
SEQ_SUM
MIN

Organization i+1
Outline

1. Introduction
   - Motivation
   - Problem Representation

2. Approach
   - Agent roles and relationships
   - Organization Formation and Adaptation

3. Progress
   - Work done till date
   - Work currently in progress
   - Future work planned
Came up with a formal definition of OSD

\[ \text{OSD}(P, \alpha) \rightarrow < O, \beta, \gamma > \]

where

- \( P = < P_1, P_2, P_3, \ldots, P_n > \)
- \( \alpha = \ldots \)
- \( O = < O_1, O_2, O_3, \ldots, O_m > \) such that \( O_i = < A_i, C_i, R_i > \)
  
  \vdots
Work done till date

- Wrote a simulator/test bed for running OSD experiments and for evaluating my approach

```lisp
(defmacro Define-Agent-Class (agent-class superclasses slots &rest options)
  `(progn
     (defclass ,agent-class
       ,(apply #'Get-Agent-Superclass-List superclasses)
       ,slots
       ,@options)
     (setf (gethash ',agent-class *Agent-Class-Info*)
           (make-instance 'AGENT-CLASS-INFO :label ',agent-class)))
)```
Work done till date

- Designed a GUI for viewing organizational changes
Evaluated our approach using 3 sets of experiments:

1. Compared with the Contract Net Protocol (CNP)
2. Evaluated the three task allocation heuristics
3. Tested the robustness of our approach

(Reported in AAMAS 2007, AOMS@IJCAI 2007, AAMAS 2006)
Work done till date
Comparison with the Contract Net Protocol
Work done till date
Evaluation of the three task allocation heuristics

Control Variables:

- Task structure depth
- Branching factor
- Probability of the CAFs
- The arrival rate
- The deadline slack
Work done till date
Evaluation of the three task allocation heuristics

Experimental Setup

- Each experiment was repeated 20 times
  - With a new randomly generated task structure
  - These 20 experiments formed an experimental set.
- A static environment was used in each experiment
- The final evaluation was done on 673 experimental sets.
Work done till date
Evaluation of the three task allocation heuristics

Experimental Setup

- Each experiment was repeated 20 times
  - With a new randomly generated task structure
  - These 20 experiments formed an experimental set.
- A static environment was used in each experiment
- The final evaluation was done on 673 experimental sets.
Experimental Setup

- Each experiment was repeated 20 times
  - With a new randomly generated task structure
  - These 20 experiments formed an *experimental set*.
- A static environment was used in each experiment
- The final evaluation was done on 673 experimental sets.
Experimental Setup

- Each experiment was repeated 20 times
  - With a new randomly generated task structure
  - These 20 experiments formed an experimental set.
- A static environment was used in each experiment
- The final evaluation was done on 673 experimental sets.
Experimental Setup

- Each experiment was repeated 20 times
  - With a new randomly generated task structure
  - These 20 experiments formed an experimental set.
- A static environment was used in each experiment
- The final evaluation was done on 673 experimental sets.
Tested Performance Criteria

1. The average number of agents used
2. The total number of organizational changes
3. The total messages sent by all the agents
4. The total resource cost of the organization
5. The number of tasks completed
6. The average quality accrued
7. The average response time of the organization
8. The average runtime of the tasks
9. The turnaround time
Work done till date
Evaluation of the three task allocation heuristics

- We ran the **Wilcoxon Matched-Pair Signed-Rank** tests on the experiments in each set.

- Null Hypothesis: *there is no difference between the pair of heuristics for the performance criteria under consideration*
  
  - Interested in the cases in which the null hypothesis can be rejected with 95% confidence ($p < 0.05$).

- We noted the number of times that a heuristic performed the best or was in a group that performed statistically better than the rest.
We ran the Wilcoxon Matched-Pair Signed-Rank tests on the experiments in each set.

Null Hypothesis: there is no difference between the pair of heuristics for the performance criteria under consideration

- Interested in the cases in which the null hypothesis can be rejected with 95% confidence ($p < 0.05$).

- We noted the number of times that a heuristic performed the best or was in a group that performed statistically better than the rest.
We ran the **Wilcoxon Matched-Pair Signed-Rank** tests on the experiments in each set.

**Null Hypothesis:** *there is no difference between the pair of heuristics for the performance criteria under consideration*

- Interested in the cases in which the null hypothesis can be rejected with 95% confidence (*p < 0.05*).

We noted the number of times that a heuristic performed the best or was in a group that performed statistically better than the rest.
Work done till date
Evaluation of the three task allocation heuristics

- We ran the **Wilcoxon Matched-Pair Signed-Rank** tests on the experiments in each set.
- **Null Hypothesis:** *there is no difference between the pair of heuristics for the performance criteria under consideration*
  - Interested in the cases in which the null hypothesis can be rejected with 95% confidence (*p* < 0.05).
- We noted the number of times that a heuristic performed the best or was in a group that performed statistically better than the rest.
Work done till date
Evaluation of the three task allocation heuristics
Figure: Robustness of the citizens approach
Outline

1. Introduction
   - Motivation
   - Problem Representation

2. Approach
   - Agent roles and relationships
   - Organization Formation and Adaptation

3. Progress
   - Work done till date
   - Work currently in progress
   - Future work planned
Work currently in progress

- Develop algorithms for and evaluate the survivalist approach [Marin et. al.] to robustness
- Investigate the effect of communication delays on the organizational structures
- Evaluate the effect of non-local interdependencies
Work currently in progress

- Develop algorithms for and evaluate the survivalist approach [Marin et. al.] to robustness
- Investigate the effect of communication delays on the organizational structures
- Evaluate the effect of non-local interdependencies
Work currently in progress

- Develop algorithms for and evaluate the survivalist approach [Marin et. al.] to robustness
- Investigate the effect of communication delays on the organizational structures
- Evaluate the effect of non-local interdependencies
Outline

1. Introduction
   - Motivation
   - Problem Representation

2. Approach
   - Agent roles and relationships
   - Organization Formation and Adaptation

3. Progress
   - Work done till date
   - Work currently in progress
   - Future work planned
Future work planned

- Evaluate the tradeoff between cloning and OSD
- Evaluate the interplay between coordination and organization
- Show how existing workflow languages might benefit from the use of our approach
- Do a case study to evaluate my approach on a practical application.
Future work planned

- Evaluate the tradeoff between cloning and OSD
- Evaluate the interplay between coordination and organization
- Show how existing workflow languages might benefit from the use of our approach
- Do a case study to evaluate my approach on a practical application.
Future work planned

- Evaluate the tradeoff between cloning and OSD
- Evaluate the interplay between coordination and organization
- Show how existing workflow languages might benefit from the use of our approach
- Do a case study to evaluate my approach on a practical application.
Future work planned

- Evaluate the tradeoff between cloning and OSD
- Evaluate the interplay between coordination and organization
- Show how existing workflow languages might benefit from the use of our approach
- Do a case study to evaluate my approach on a practical application.
Conclusion

How many agents does it take to screw in a light bulb?

Answer: It’s an open research question ...
How many agents does it take to screw in a light bulb?

Answer: It’s an open research question ...
Questions?
Reasons for Organizational Change

- Organizational change is expensive
  - requires clock cycles
  - allocation/deallocation of resources
  - various other overhead

- Hence, stable organizations are desirable
- Therefore, only change the organizational structure if
  - the task structure changes; or
  - the environmental conditions change

- Right now, we only handle the latter
for Task$_i$ in OutstandingTaskQueue do
    Compute $t_{avail_i}$, the time available for Task$_i$
    if $t_{avail_i} < g_{min}(Task_i)$ then
        SpawnAgent()
    else
        if $g_{min}(Task_i) < t_{avail_i} < expt(Task_i)$ then
            SpawnAgent() with probability p
        else
            Do Nothing
        end if
    end if
end for
Computing Time Available

Task 1

Task 2

Task 3

Task 4
Computing Time Available

- Divide total time into regions

![Diagram showing time regions and tasks]

- Task 1
- Task 2
- Task 3
- Task 4

Time regions: R1, R2, R3, R4, R5
Computing Time Available

- Compute *time available per task* for each region

\[ t_i = \frac{2}{4} \]
Computing Time Available

- For each task, sum up the *time available per task* for each region in which the task appears.
Reasons for Organizational Change
Detecting Agent Overload

\[
\begin{align*}
&\text{for } Task_i \text{ in } OutstandingTaskQueue \text{ do} \\
&\quad \text{Compute } t_{avail_i}, \text{ the time available for } Task_i \\
&\quad \text{\textbf{if }} t_{avail_i} < gmin(Task_i) \text{ then} \\
&\quad\quad \text{SpawnAgent()} \\
&\quad \text{\textbf{else}} \\
&\quad\quad \text{\textbf{if }} gmin(Task_i) < t_{avail_i} < \text{expt}(Task_i) \text{ then} \\
&\quad\quad\quad \text{SpawnAgent()} \text{ with probability } p \\
&\quad\quad \text{\textbf{else}} \\
&\quad\quad\quad \text{Do Nothing} \\
&\quad \text{\textbf{end if}} \\
&\text{\textbf{end if}} \\
&\text{end for}
\end{align*}
\]
Reasons for Organizational Change

- Furthermore, change organizational structure only if the change is permanent
  - Impossible to know
- Make the probability of change be inversely proportional to the time since the last organizational change
  - If time since last change is relatively short, the agents are still adjusting to the changes in the environment
    - the probability of an agent initiating an organizational change should be high.
  - if the time since the last organizational change is relatively large, we have a well-suited stable organizational structure
    - the probability of an agent initiating an organizational change should be low.
- To allow this variation, we use Simulated Annealing.
Reasons for Organizational Change

- Furthermore, change organizational structure only if the change is permanent
  - Impossible to know

- Make the probability of change be inversely proportional to the time since the last organizational change
  - If time since last change is relatively short, the agents are still adjusting to the changes in the environment
    - the probability of an agent initiating an organizational change should be high.
  - if the time since the last organizational change is relatively large, we have a well-suited stable organizational structure
    - the probability of an agent initiating an organizational change should be low.

- To allow this variation, we use Simulated Annealing.
Furthermore, change organizational structure only if the change is permanent.

Impossible to know

Make the probability of change be inversely proportional to the time since the last organizational change.

If time since last change is relatively short, the agents are still adjusting to the changes in the environment, the probability of an agent initiating an organizational change should be high.

If the time since the last organizational change is relatively large, we have a well-suited stable organizational structure, the probability of an agent initiating an organizational change should be low.

To allow this variation, we use Simulated Annealing.
Reasons for Organizational Change

- Furthermore, change organizational structure only if the change is permanent
  - Impossible to know

- Make the probability of change be inversely proportional to the time since the last organizational change
  - If time since last change is relatively short, the agents are still adjusting to the changes in the environment
    - The probability of an agent initiating an organizational change should be high.
  - If the time since the last organizational change is relatively large, we have a well-suited stable organizational structure
    - The probability of an agent initiating an organizational change should be low.

- To allow this variation, we use Simulated Annealing.
Reasons for Organizational Change

- Furthermore, change organizational structure only if the change is permanent
  - Impossible to know
- Make the probability of change be inversely proportional to the time since the last organizational change
  - If time since last change is relatively short, the agents are still adjusting to the changes in the environment
    - the probability of an agent initiating an organizational change should be high.
  - if the time since the last organizational change is relatively large, we have a well-suited stable organizational structure
    - the probability of an agent initiating an organizational change should be low.
- To allow this variation, we use Simulated Annealing.
Reasons for Organizational Change

- Furthermore, change organizational structure only if the change is permanent
  - Impossible to know

- Make the probability of change be inversely proportional to the time since the last organizational change
  - If time since last change is relatively short, the agents are still adjusting to the changes in the environment
    - the probability of an agent initiating an organizational change should be high.
  - if the time since the last organizational change is relatively large, we have a well-suited stable organizational structure
    - the probability of an agent initiating an organizational change should be low.

- To allow this variation, we use Simulated Annealing.
Reasons for Organizational Change

- Furthermore, change organizational structure only if the change is permanent
  - Impossible to know

- Make the probability of change be inversely proportional to the time since the last organizational change
  - If time since last change is relatively short, the agents are still adjusting to the changes in the environment
    - the probability of an agent initiating an organizational change should be high.
  - if the time since the last organizational change is relatively large, we have a well-suited stable organizational structure
    - the probability of an agent initiating an organizational change should be low.

- To allow this variation, we use Simulated Annealing.
Furthermore, change organizational structure only if the change is permanent

Impossible to know

Make the probability of change be inversely proportional to the time since the last organizational change

If time since last change is relatively short, the agents are still adjusting to the changes in the environment

the probability of an agent initiating an organizational change should be high.

if the time since the last organizational change is relatively large, we have a well-suited stable organizational structure

the probability of an agent initiating an organizational change should be low.

To allow this variation, we use Simulated Annealing.
Reasons for Organizational Change

Simulated Annealing

- The probability of keeping an existing organizational structure is calculated using the annealing formula,
  \[ p = e^{-\frac{\Delta E}{kT}}, \]
  where
  - \( \Delta E \) is the “amount” of overload/underload,
  - \( T \) is the time since the last organizational change; and
  - \( k \) is a constant.

- If \( T \) is large, \( p \), or the probability of keeping the existing organizational structure is large.
  - Problem: The organization may be too sluggish in its reaction to organizational change or may not respond at all.
  - Solution: Cap \( p \) at a certain threshold.

- The probability of organizational change, \( q = 1 - p \)

- The mechanism of computing \( \Delta E \) is different for agent spawning than for agent composition.
Simulated Annealing

- The probability of keeping an existing organizational structure is calculated using the annealing formula,

\[ p = e^{-\frac{\Delta E}{kT}}, \text{ where} \]

- \( \Delta E \) is the "amount" of overload/underload,
- \( T \) is the time since the last organizational change; and
- \( k \) is a constant.

- If \( T \) is large, \( p \), or the probability of keeping the existing organizational structure is large.
  - Problem: The organization may be too sluggish in its reaction to organizational change or may not respond at all.
  - Solution: Cap \( p \) at a certain threshold

- The probability of organizational change, \( q = 1 - p \)
- The mechanism of computing \( \Delta E \) is different for agent spawning than for agent composition
Reasons for Organizational Change

Simulated Annealing

- The probability of keeping an existing organizational structure is calculated using the annealing formula,
  \[ p = e^{-\frac{\Delta E}{kT}}, \]
  where
  - \( \Delta E \) is the "amount" of overload/underload,
  - \( T \) is the time since the last organizational change; and
  - \( k \) is a constant.

- If \( T \) is large, \( p \), or the probability of keeping the existing organizational structure is large.
  - Problem: The organization may be too sluggish in its reaction to organizational change or may not respond at all.
  - Solution: Cap \( p \) at a certain threshold

- The probability of organizational change, \( q = 1 - p \)
- The mechanism of computing \( \Delta E \) is different for agent spawning than for agent composition.
Reasons for Organizational Change

Simulated Annealing

- The probability of keeping an existing organizational structure is calculated using the annealing formula,
\[ p = e^{-\frac{\Delta E}{kT}}, \] where
  - \( \Delta E \) is the "amount" of overload/underload,
  - \( T \) is the time since the last organizational change; and
  - \( k \) is a constant.
  - If \( T \) is large, \( p \), or the probability of keeping the existing organizational structure is large.
    - Problem: The organization may be too sluggish in its reaction to organizational change or may not respond at all.
    - Solution: Cap \( p \) at a certain threshold
  - The probability of organizational change, \( q = 1 - p \)
  - The mechanism of computing \( \Delta E \) is different for agent spawning than for agent composition
Simulated Annealing

- The probability of keeping an existing organizational structure is calculated using the annealing formula,

\[ p = e^{-\frac{\Delta E}{kT}}, \]

where

- \( \Delta E \) is the “amount” of overload/underload,
- \( T \) is the time since the last organizational change; and
- \( k \) is a constant.

- If \( T \) is large, \( p \), or the probability of keeping the existing organizational structure is large.

  - Problem: The organization may be too sluggish in its reaction to organizational change or may not respond at all.
  - Solution: Cap \( p \) at a certain threshold

- The probability of organizational change, \( q = 1 - p \)
- The mechanism of computing \( \Delta E \) is different for agent spawning than for agent composition

Reasons for Organizational Change

Simulated Annealing

- The probability of keeping an existing organizational structure is calculated using the annealing formula,

\[ p = e^{-\frac{\Delta E}{kT}}, \]

where

- \( \Delta E \) is the “amount” of overload/underload,
- \( T \) is the time since the last organizational change; and
- \( k \) is a constant.

- If \( T \) is large, \( p \), or the probability of keeping the existing organizational structure is large.

  - Problem: The organization may be too sluggish in its reaction to organizational change or may not respond at all.
  - Solution: Cap \( p \) at a certain threshold

- The probability of organizational change, \( q = 1 - p \)
- The mechanism of computing \( \Delta E \) is different for agent spawning than for agent composition
Reasons for Organizational Change

Simulated Annealing

- The probability of keeping an existing organizational structure is calculated using the annealing formula, 
  
  \[ p = e^{-\frac{\Delta E}{kT}} \] , where

  - \( \Delta E \) is the “amount” of overload/underload,
  - \( T \) is the time since the last organizational change; and
  - \( k \) is a constant.

- If \( T \) is large, \( p \), or the probability of keeping the existing organizational structure is large.
  
  - Problem: The organization may be too sluggish in its reaction to organizational change or may not respond at all.

  - Solution: Cap \( p \) at a certain threshold

- The probability of organizational change, \( q = 1 - p \)

- The mechanism of computing \( \Delta E \) is different for agent spawning than for agent composition
Reasons for Organizational Change

Simulated Annealing

- The probability of keeping an existing organizational structure is calculated using the annealing formula,
  \[ p = e^{-\frac{\Delta E}{kT}} \], where
  - \( \Delta E \) is the "amount" of overload/underload,
  - \( T \) is the time since the last organizational change; and
  - \( k \) is a constant.

- If \( T \) is large, \( p \), or the probability of keeping the existing organizational structure is large.
  - Problem: The organization may be too sluggish in its reaction to organizational change or may not respond at all.
  - Solution: Cap \( p \) at a certain threshold

- The probability of organizational change, \( q = 1 - p \)
- The mechanism of computing \( \Delta E \) is different for agent spawning than for agent composition.
Simulated Annealing

- The probability of keeping an existing organizational structure is calculated using the annealing formula,

\[ p = e^{-\frac{\Delta E}{kT}} \],

where

- \( \Delta E \) is the “amount” of overload/underload,
- \( T \) is the time since the last organizational change; and
- \( k \) is a constant.

- If \( T \) is large, \( p \), or the probability of keeping the existing organizational structure is large.
  - Problem: The organization may be too sluggish in its reaction to organizational change or may not respond at all.
  - Solution: Cap \( p \) at a certain threshold

- The probability of organizational change, \( q = 1 - p \)

- The mechanism of computing \( \Delta E \) is different for agent spawning than for agent composition
Reasons for Organizational Change

Computation of $\Delta E$ for Agent Spawning

To compute if spawning is necessary, $\Delta E = \frac{1}{\alpha \times \text{Slack}}$, where

- $\alpha$ is a constant
- $\text{Slack}$ is the difference between the total time available for completion of the outstanding tasks and the sum of the expected time required for completion of each task on the task queue
Computation of $\Delta E$ for Agent Composition

- To calculate if agent composition is necessary, $\Delta E = \beta \times \text{Idle\_Time}$, where
  - $\beta$ is a constant
  - $\text{Idle\_Time}$ is the amount of time for which the agent was idle

- If the agent has been sitting idle for a long period of time, $\Delta E$ is large, which implies that $p$, the probability of keeping the existing organizational structure, is low.
Two commonly used approaches to achieve robustness in multiagent systems are:

1. **Survivalist Approach** [Marin et. al.]
   - Involves replicating domain agents
   - Replicas can take over should the original agents fail
   - Advantage: fast recovery

2. **Citizen Approach** [Dellarocas et. al.]
   - Involves the use of special monitoring agents (called *Sentinel Agents*)
   - Sentinels detect agent failure and dynamically startup new agents in lieu of the failed ones
   - Advantage: simpler to implement
   - Advantage: requires little modification to the existing organizational structure and coordination protocol.
Both of these approaches can be applied to OSD

We chose the citizens approach

- shown by [Dellarocas et. al.] to have better performance in the CNP.

We designed special monitoring agents

- that send periodic “are you alive” messages to the domain agents
- If domain agent fails, it won’t respond to these polls
- After timeout, monitoring agent creates a new agent and delegates the responsibilities of the failed agent to it.

Delegation of responsibilities to new agent is non-trivial

- requires the deducting of state information from a record of the messages received by the failed agent
Experimental Setup

- Ran 40 experiments:
  - 20 experiments had a static environment
    - Arrival rate: 15 cycles (fixed)
    - Deadline window: 20 cycles (fixed)
  - 20 experiments had a dynamic environment
    - Arrival rate: varied from 15 cycles to 30 and back to 15 after every 20 tasks.
    - Deadline window: 20 cycles (fixed)

- All task structures were randomly generated
  - Maximum depth: 4
  - Maximum branching factor: 3

- Each experiment was repeated 5 times:
  - OSD approach was used the first time
  - CNP approach with 8, 10, 12, 14 agents were used in the subsequent 4 runs

- The runtime of all the experiments was 2500 cycles.
Work done till date
Comparison with the contract net protocol — static environments

Figure: Average performance in static environments
Work done till date
Comparison with the contract net protocol — static environments

Figure: Performance of individual exp. in a static environment
Figure: Average performance in dynamic environments
Work done till date
Comparison with the contract net protocol — dynamic environments

**Figure:** Performance of individual exp. in dynamic environment