

Organizational Self-Design in Semi-Dynamic Environments

Sachin Kamboj and Keith S. Decker

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

Sixth International Joint Conference on Autonomous Agents
and Multiagent Systems - (AAMAS 2007)

Outline

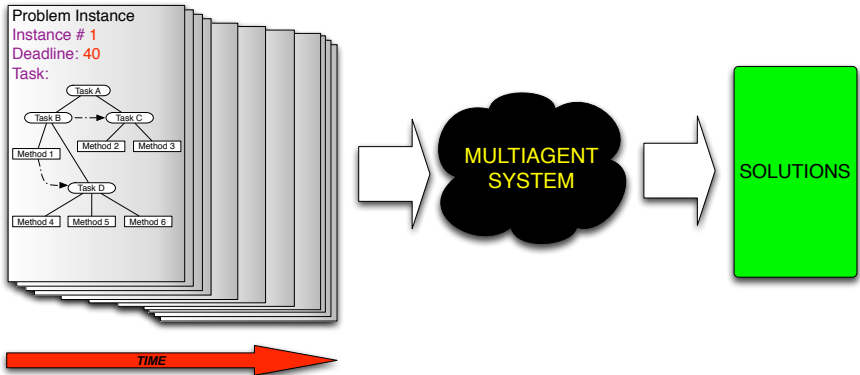
- 1 Introduction
 - Motivation
 - Problem Representation
- 2 Approach
 - Agent roles and relationships
 - Organization Formation and Adaptation
- 3 Evaluation
 - Types of Experiments
 - Comparison with the Contract Net Protocol
 - Evaluation of the three task allocation heuristics

Outline

- 1 Introduction
 - Motivation
 - Problem Representation
- 2 Approach
 - Agent roles and relationships
 - Organization Formation and Adaptation
- 3 Evaluation
 - Types of Experiments
 - Comparison with the Contract Net Protocol
 - Evaluation of the three task allocation heuristics

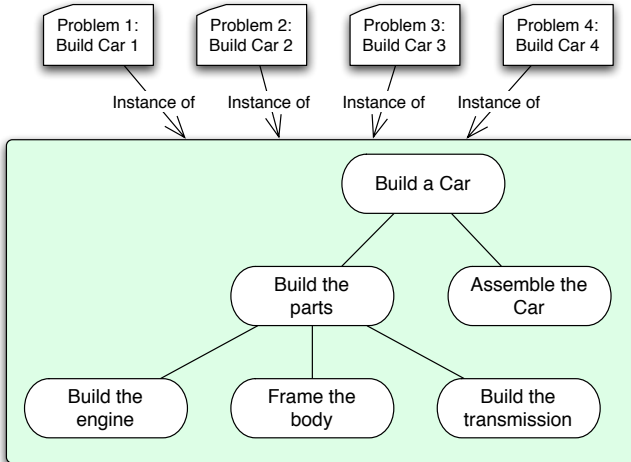
Introduction

Basic Model



Introduction

Problems



Introduction

Problems

Environmental conditions

- Problems arrive at varying rates
- Problems might have different deadlines
- Available resources might change over time

Introduction

Problems

Environmental conditions

- Problems arrive at varying rates
- Problems might have different deadlines
- Available resources might change over time

Introduction

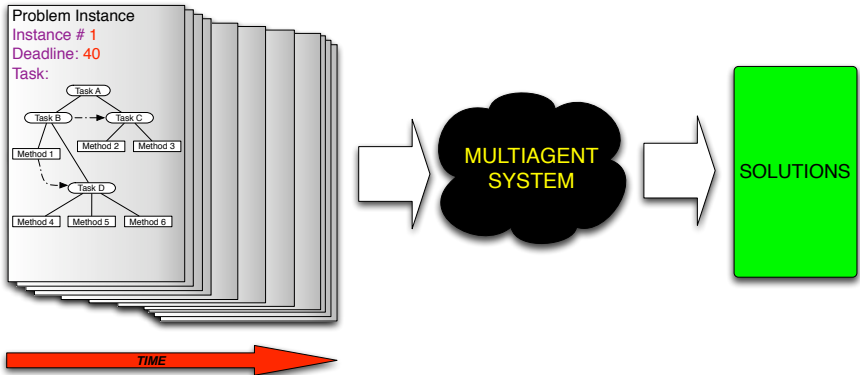
Problems

Environmental conditions

- Problems arrive at varying rates
- Problems might have different deadlines
- Available resources might change over time

Introduction

Basic Model



Introduction

Motivation

Organizational Issues

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

Introduction

Motivation

Organizational Issues

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

Introduction

Motivation

Organizational Issues

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

Introduction

Motivation

So what's the problem?

- There is no best way of organizing and all ways of organizing are not equally effective
- Environmental may be dynamic
- All problem instances and environmental conditions are not unique

Introduction

Motivation

So what's the problem?

- There is no best way of organizing and all ways of organizing are not equally effective
- Environmental may be dynamic
- All problem instances and environmental conditions are not unique

Introduction

Motivation

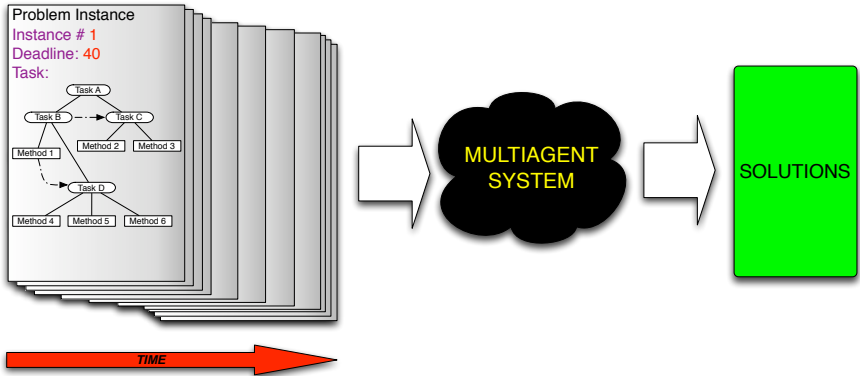
So what's the problem?

- There is no best way of organizing and all ways of organizing are not equally effective
- Environmental may be dynamic
- All problem instances and environmental conditions are not unique

Introduction

Basic Model

Our Answer:



Introduction

Motivation

Our Answer:

We propose to use a dynamic run-time approach to organization based on Organizational-Self Design (OSD) [Ishida, Gasser, et al.]

- OSD means agents design their own organizational structures
- *Our main contribution is extending existing OSD research to incorporate TÆMS based task structures.*

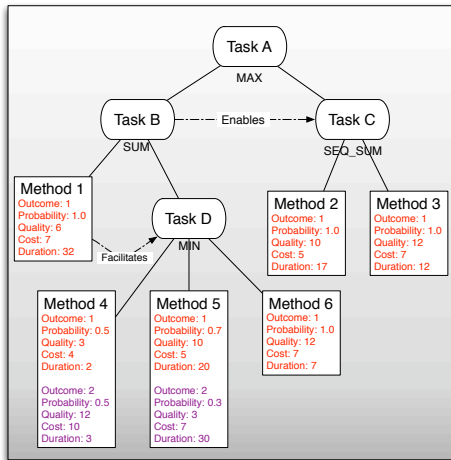
Outline

- 1 Introduction
 - Motivation
 - Problem Representation
- 2 Approach
 - Agent roles and relationships
 - Organization Formation and Adaptation
- 3 Evaluation
 - Types of Experiments
 - Comparison with the Contract Net Protocol
 - Evaluation of the three task allocation heuristics

Introduction

Problem Representation

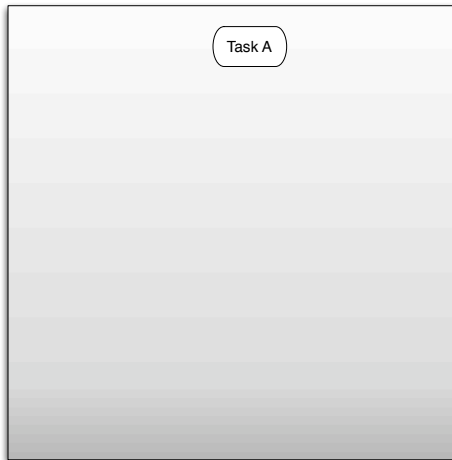
Problems are represented using TÆEMS



Introduction

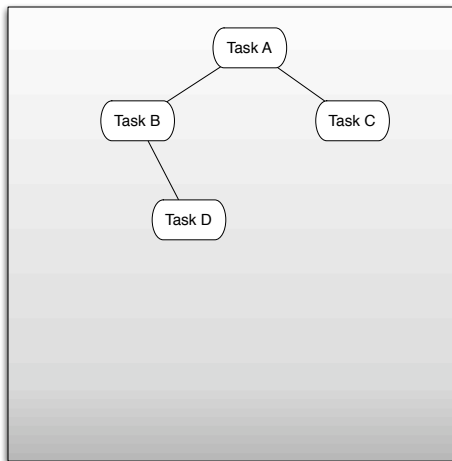
Problem Representation

Problems are represented using TÆEMS



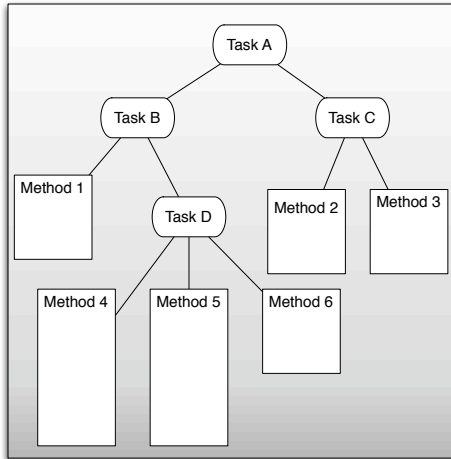
Introduction

Problem Representation



Introduction

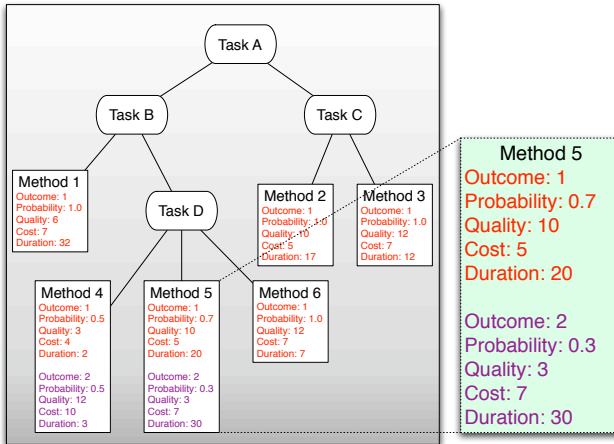
Problem Representation



Introduction

Problem Representation

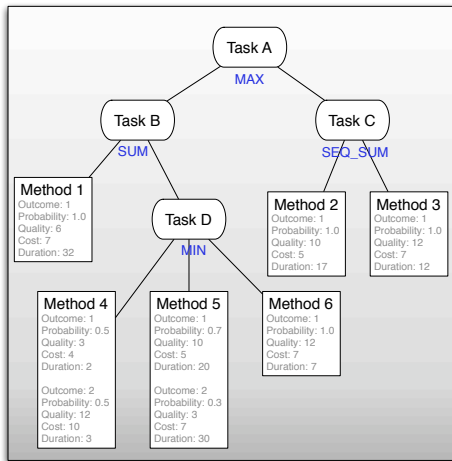
TÆMS is quantitative



Introduction

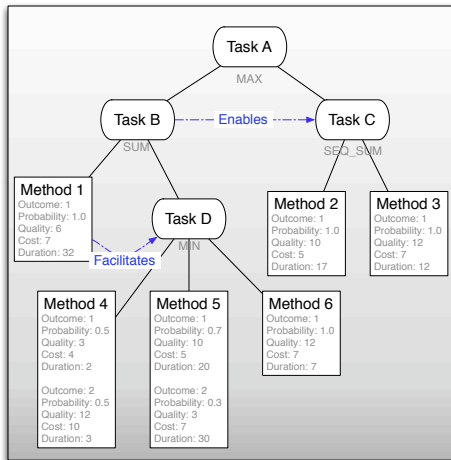
Problem Representation

TÆEMS is quantitative



Introduction

Problem Representation



Outline

- 1 Introduction
 - Motivation
 - Problem Representation
- 2 Approach
 - **Agent roles and relationships**
 - Organization Formation and Adaptation
- 3 Evaluation
 - Types of Experiments
 - Comparison with the Contract Net Protocol
 - Evaluation of the three task allocation heuristics

Organizational Design

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 TÆEMS 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 TÆEMS 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 TÆEMS subtree rooted at a particular node

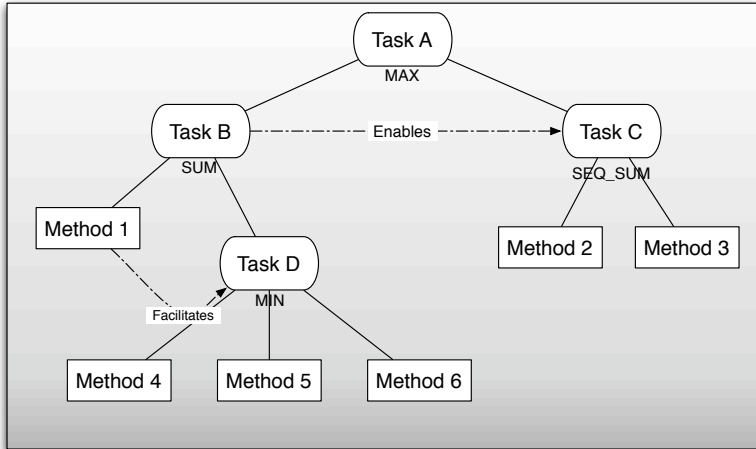
Agent roles and relationships

- 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

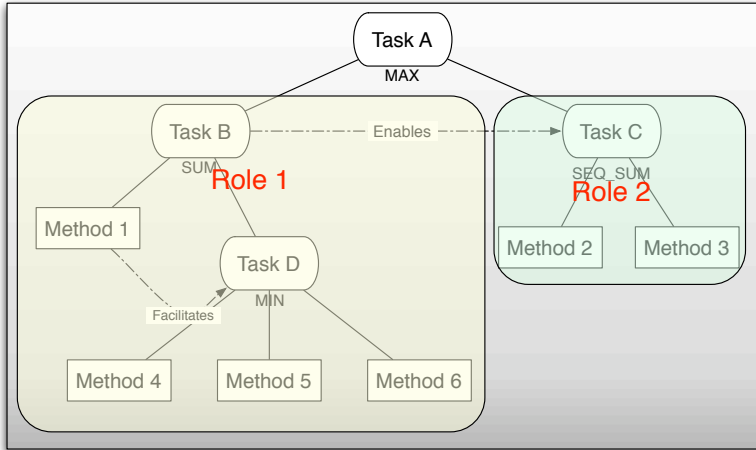
Definition

A role is a TÆEMS subtree rooted at a particular node

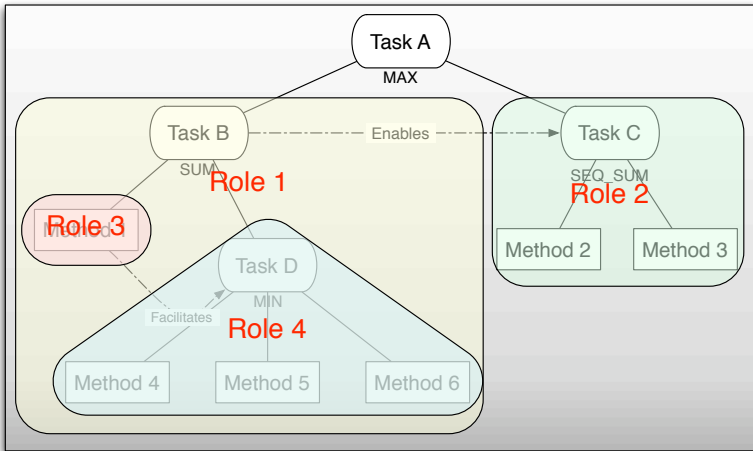
Agent roles and relationships



Agent roles and relationships



Agent roles and relationships



Outline

- 1 Introduction
 - Motivation
 - Problem Representation
- 2 Approach
 - Agent roles and relationships
 - **Organization Formation and Adaptation**
- 3 Evaluation
 - Types of Experiments
 - Comparison with the Contract Net Protocol
 - Evaluation of the three task allocation heuristics

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 requests that an agent be added to the organization
 - It is free (underloaded)
 - It requests that a suitable agent be added to the organization

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

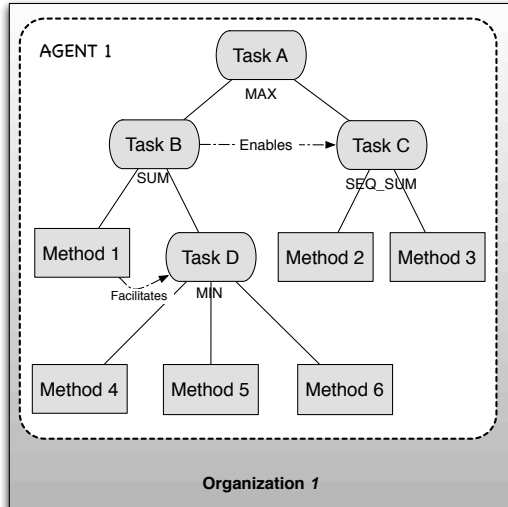
- Start off with an initial organization consisting of a single agent, solely responsible for all activities
- Each agent in the organization checks to see if:
 - It is overloaded:
 - It spawns off a new agent to handle part of its load
 - It is free (underloaded)
 - It combines with another agent to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

Organization Formation and Adaptation

- 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

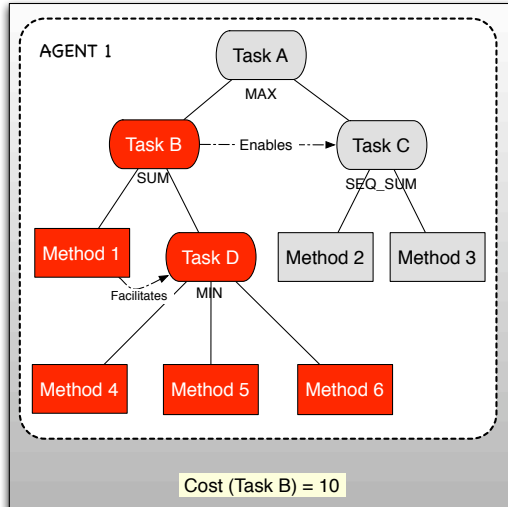
Organization Formation and Adaptation

Agent Spawning



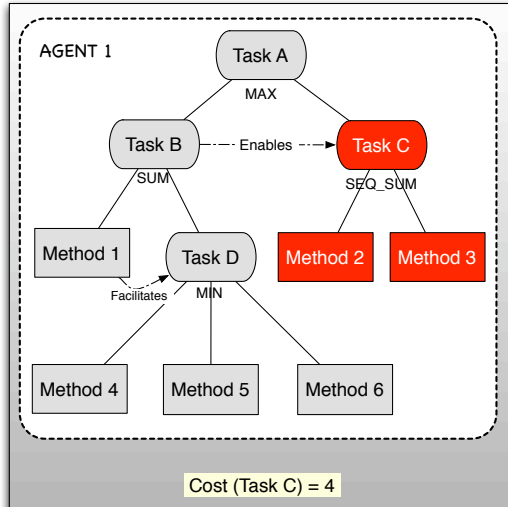
Organization Formation and Adaptation

Agent Spawning



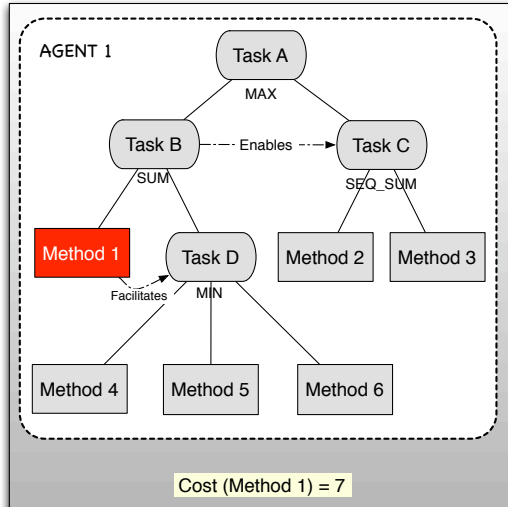
Organization Formation and Adaptation

Agent Spawning



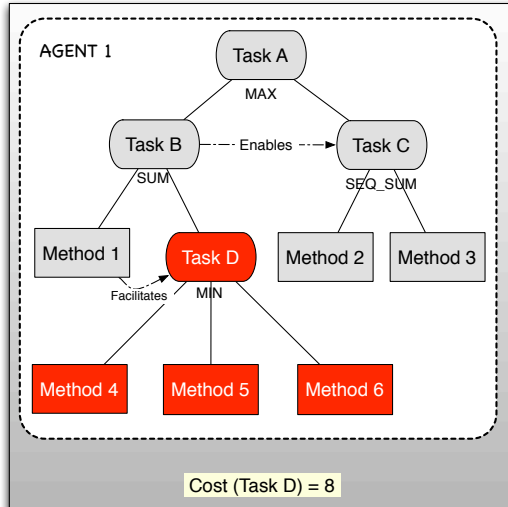
Organization Formation and Adaptation

Agent Spawning



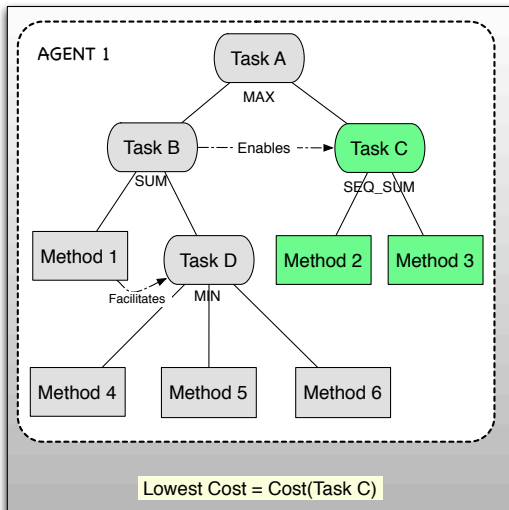
Organization Formation and Adaptation

Agent Spawning



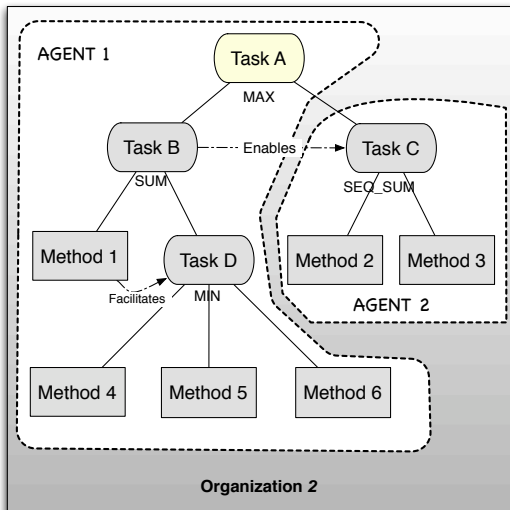
Organization Formation and Adaptation

Agent Spawning



Organization Formation and Adaptation

Agent Spawning



Organization Formation and Adaptation

Agent Spawning

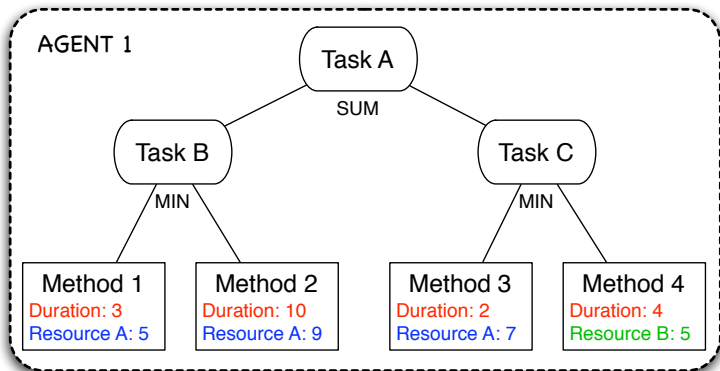
What is the cost function?

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

Organization Formation and Adaptation

Agent Spawning

The three cost functions

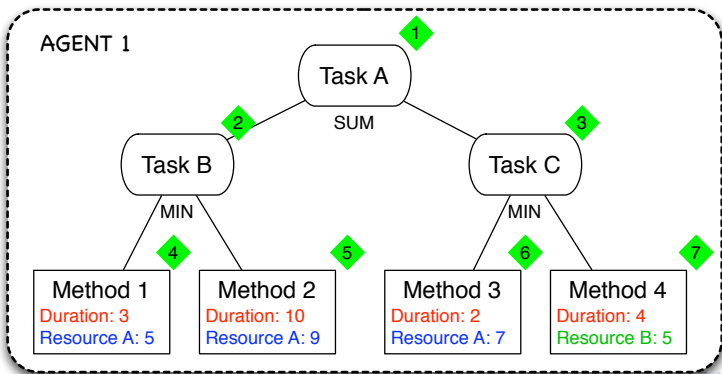


Organization Formation and Adaptation

Agent Spawning

Topmost First

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

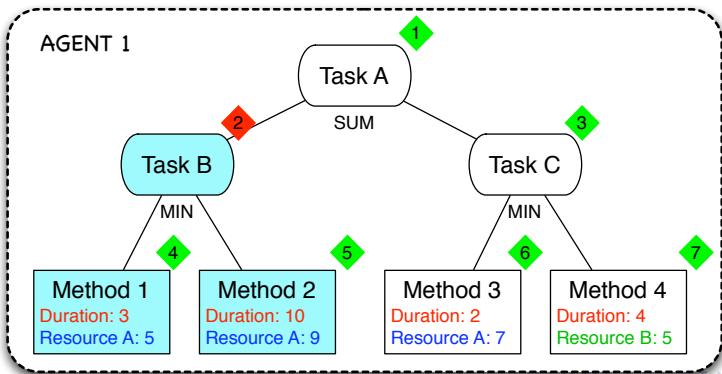


Organization Formation and Adaptation

Agent Spawning

Topmost First

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

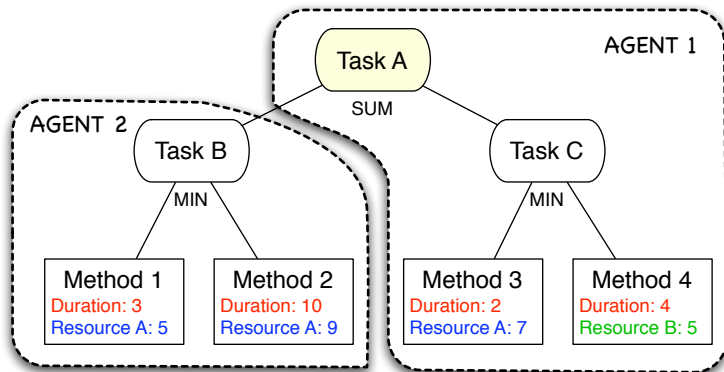


Organization Formation and Adaptation

Agent Spawning

Topmost First

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

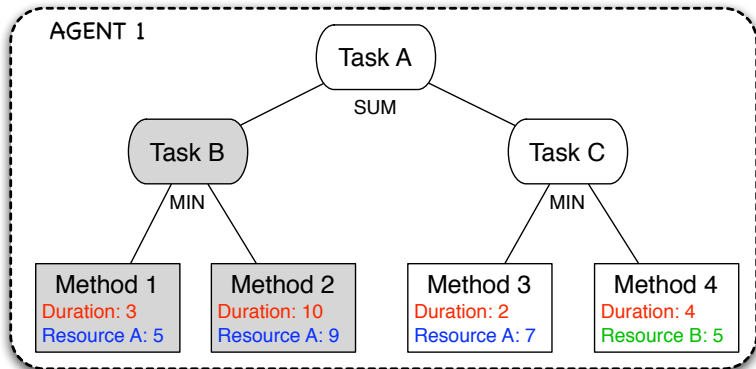


Organization Formation and Adaptation

Agent Spawning

Minimizing Resource Cost

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

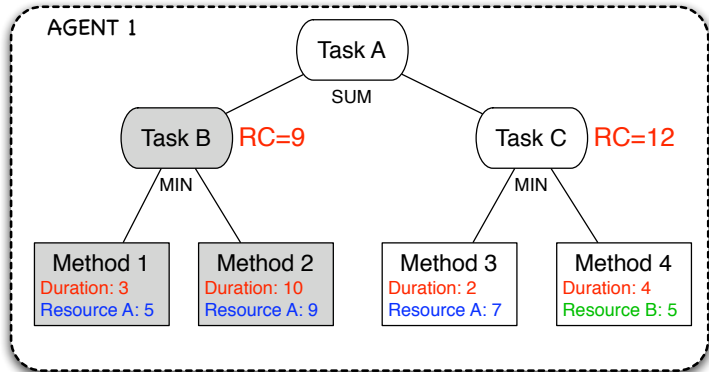


Organization Formation and Adaptation

Agent Spawning

Minimizing Resource Cost

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

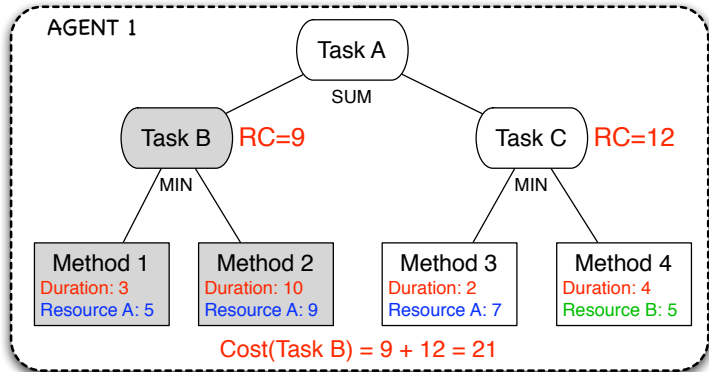


Organization Formation and Adaptation

Agent Spawning

Minimizing Resource Cost

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

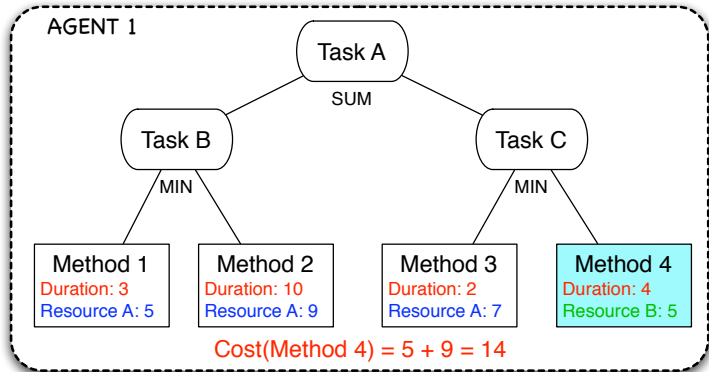


Organization Formation and Adaptation

Agent Spawning

Minimizing Resource Cost

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

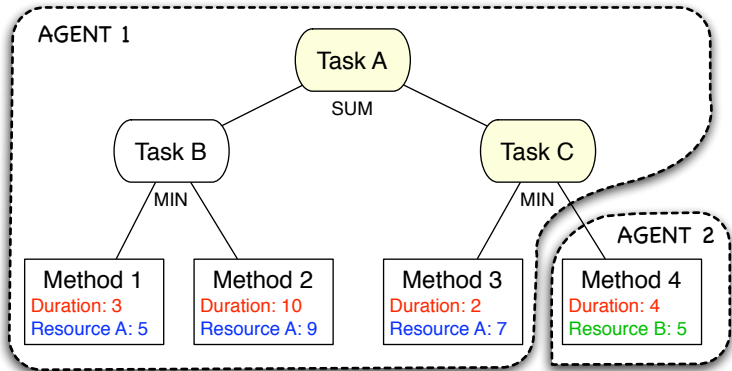


Organization Formation and Adaptation

Agent Spawning

Minimizing Resource Cost

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

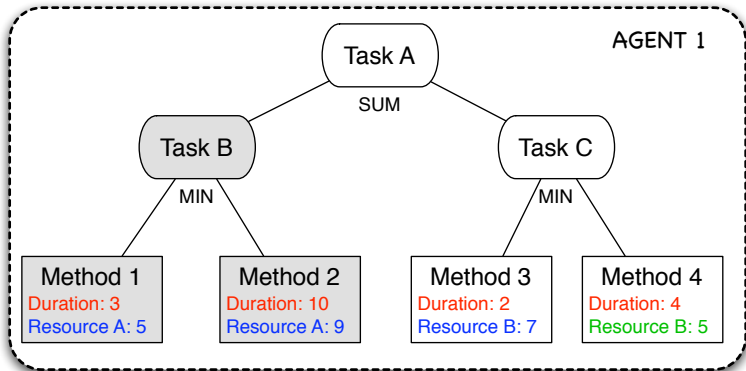


Organization Formation and Adaptation

Agent Spawning

Balancing Execution Time

- $\text{Cost}(\text{Role}) = | \text{Expt Duration}(\text{Role}) - \text{Expt Duration}(\overline{\text{Role}}) |$

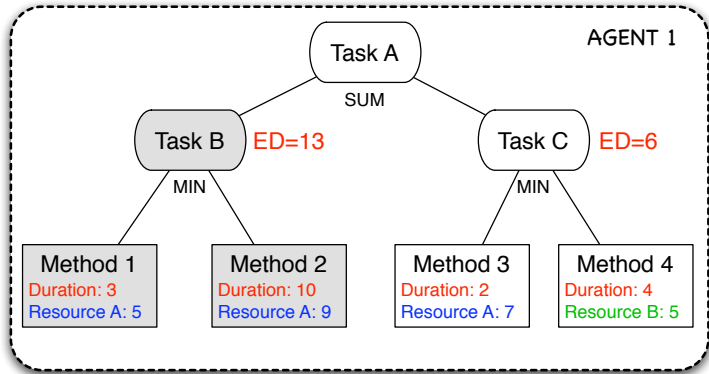


Organization Formation and Adaptation

Agent Spawning

Balancing Execution Time

- $\text{Cost}(\text{Role}) = | \text{Expt Duration}(\text{Role}) - \text{Expt Duration}(\overline{\text{Role}}) |$

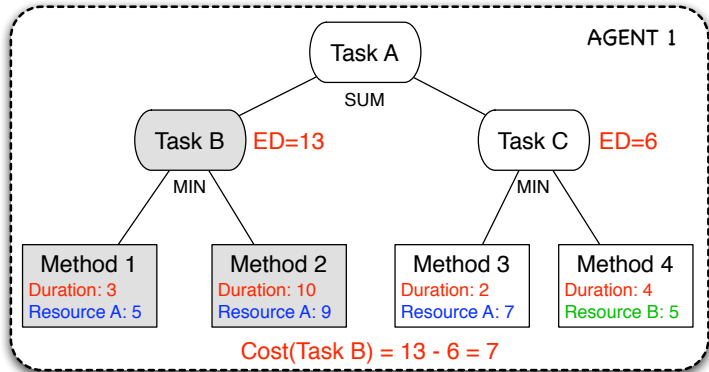


Organization Formation and Adaptation

Agent Spawning

Balancing Execution Time

- $Cost(Role) = | \text{Expt Duration}(Role) - \text{Expt Duration}(\overline{Role}) |$

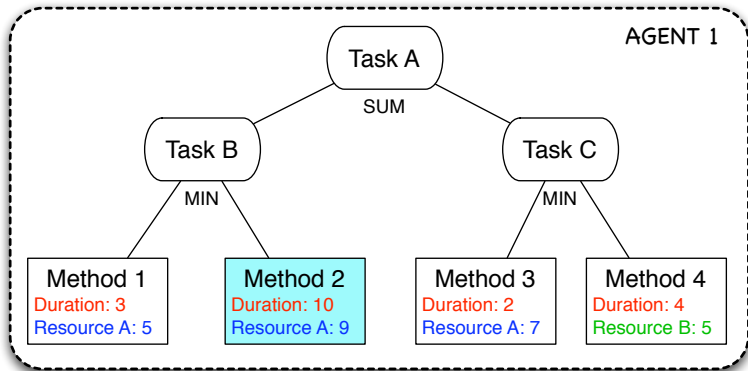


Organization Formation and Adaptation

Agent Spawning

Balancing Execution Time

- $\text{Cost}(\text{Role}) = | \text{Expt Duration}(\text{Role}) - \text{Expt Duration}(\overline{\text{Role}}) |$

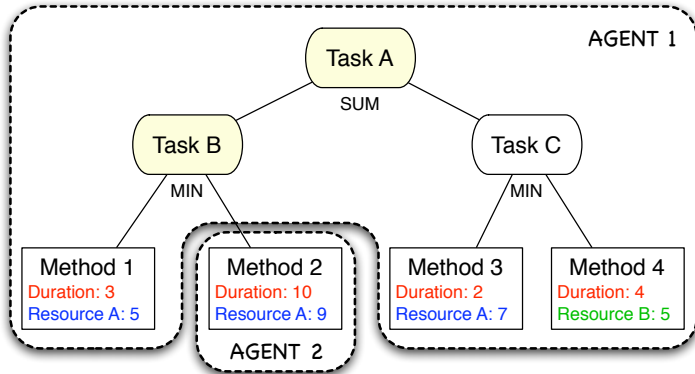


Organization Formation and Adaptation

Agent Spawning

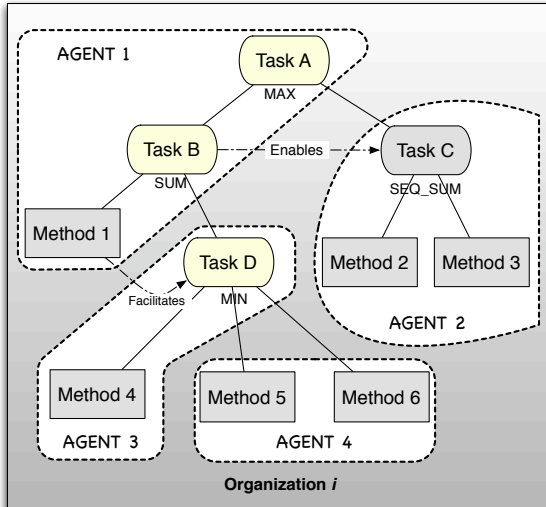
Balancing Execution Time

- $\text{Cost}(\text{Role}) = | \text{Expt Duration}(\text{Role}) - \text{Expt Duration}(\overline{\text{Role}}) |$



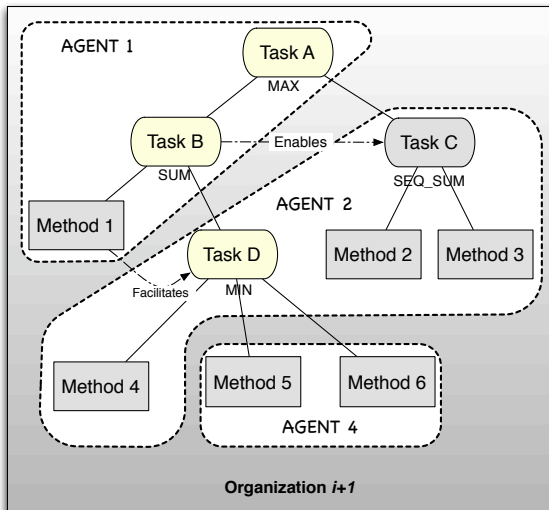
Organization Formation and Adaptation

Agent Composition



Organization Formation and Adaptation

Agent Composition



Outline

- 1 Introduction
 - Motivation
 - Problem Representation
- 2 Approach
 - Agent roles and relationships
 - Organization Formation and Adaptation
- 3 Evaluation
 - **Types of Experiments**
 - Comparison with the Contract Net Protocol
 - Evaluation of the three task allocation heuristics

Evaluation and Results

We conducted 3 sets of experiments to evaluate our approach:

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

Evaluation and Results

We conducted 3 sets of experiments to evaluate our approach:

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

Evaluation and Results

We conducted 3 sets of experiments to evaluate our approach:

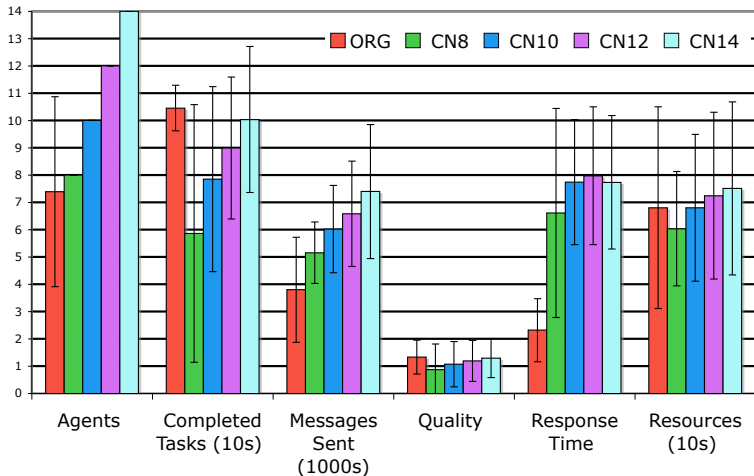
- 1 Compared with the Contract Net Protocol (CNP)
- 2 Evaluated the three task allocation heuristics
- 3 Tested the robustness of our approach

Outline

- 1 Introduction
 - Motivation
 - Problem Representation
- 2 Approach
 - Agent roles and relationships
 - Organization Formation and Adaptation
- 3 Evaluation
 - Types of Experiments
 - **Comparison with the Contract Net Protocol**
 - Evaluation of the three task allocation heuristics

Evaluation

Comparison with the Contract Net Protocol



Outline

- 1 Introduction
 - Motivation
 - Problem Representation
- 2 Approach
 - Agent roles and relationships
 - Organization Formation and Adaptation
- 3 Evaluation
 - Types of Experiments
 - Comparison with the Contract Net Protocol
 - Evaluation of the three task allocation heuristics

Evaluation

Evaluation of the three task allocation heuristics

Control Variables:

- Task structure depth
- Branching factor
- Probability of the CAFs
- The arrival rate
- The deadline slack

Evaluation

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.

Evaluation

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.

Evaluation

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.

Evaluation

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.

Evaluation

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.

Evaluation

Evaluation of the three task allocation heuristics

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

Evaluation

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.

Evaluation

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.

Evaluation

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.

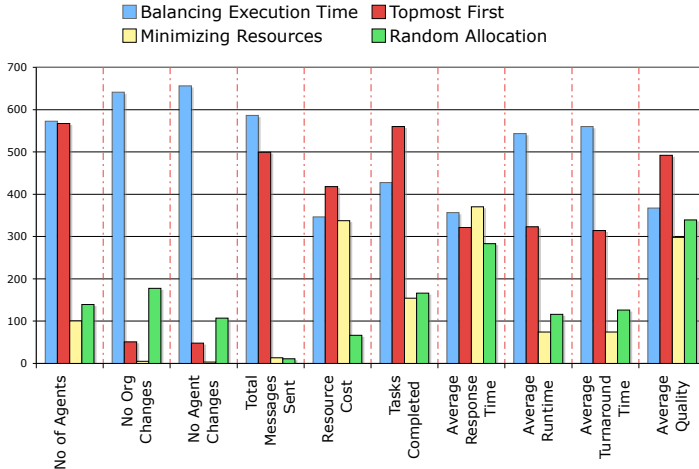
Evaluation

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.

Evaluation

Evaluation of the three task allocation heuristics



Questions?

