Autonomy Oriented Computing (AOC) for Web Intelligence (WI): A Distributed Resource Optimization Perspective

Xiaolong Jin

Principal-Supervisor: Prof. Jiming Liu

Co-supervisor: Prof. Yuan-Yan Tang

Outline

- Introduction
  - The Web and its impact
  - The proposal of Web Intelligence (WI)
- Related work
  - Web Intelligence (WI)
  - Autonomy Oriented Computing (AOC)
- Research problems

Outline (Cont.)

- Distributed resource optimization (DRO) perspective on WI
  - A DRO perspective and a generalized DRO scenario
  - WI requirements on DRO
- An AOC formulation for DRO
- DRO in homogeneous environments
  - AOC-based DRO paradigm
  - AOC-based instantaneous DRO model
  - AOC-based ongoing DRO model
  - Experimental validation
  - Summary

Outline (Cont.)

- DRO in homogeneous environments
  - Characterization of heterogeneous resource environments
  - AOC-based DRO paradigm
  - Experimental validation
  - Summary
- Significance and contribution
- Future work

Introduction

The Web and Its Impacts

- Has fast and broadly entered people's daily life
  - Astronomical computer hosts, websites, and Internet users
- Has been making strong and profound impacts on our conventional ways of working, living, learning, and playing. The Web has been widely utilized in:
  - Getting information
  - Entertainment
  - Study
  - Communication
  - Making friends
  - …

The Proposal of WI

- Proposal: by **Zhong, Liu, Yao,** and **Ohsuga**; in COMPSAC 2000
- Purpose: To explore the advanced Web technologies and corresponding theories for the next generation of the Web
- An informal definition:

> "Broadly speaking, Web Intelligence (WI) is a new direction for scientific research and development that explores the *fundamental roles* as well as *practical impacts* of *Artificial Intelligence (AI)* and *advanced Information Technology (IT)* on the next generation of *Web-empowered products, systems, services, and activities*. It is the key and the most urgent research field of IT in the era of Web and agent intelligence."

Related Work

Why WI?

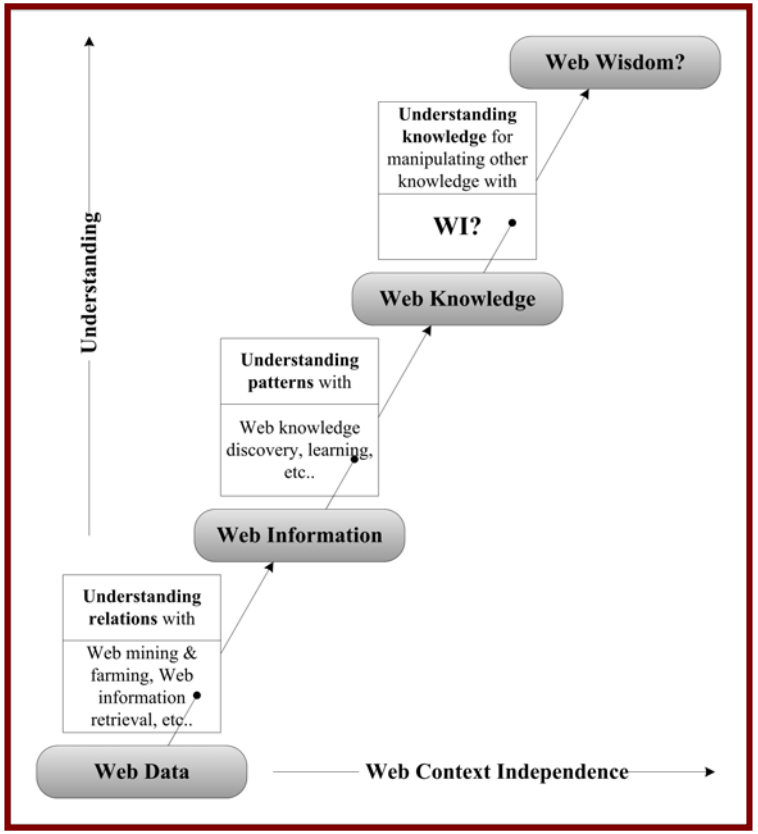
- The Web puts forward a lot of *new research issues* to artificial intelligence (AI) theories and information technologies (ITs)
- Some *old artificial intelligence (AI) or intelligent agent technology (IAT) related research issues* also exist in the Web
- Many companies concerns how to provide intelligent solutions to their Web-based or Web-supported business

The Goal of WI

- Questions:
  - How to make revolutionary innovation on the Web?
  - What will be the next paradigm shift of the Web?
- Answer: the **Wisdom Web**
  - Proposal: by **Liu, Zhong, Yao**, and **Ras**; in 2002
  - Purpose: enable human users to gain new practical *wisdom* of working, living, learning, and playing
  - How to:
    - Enable the Web with the capability to make due use of Web knowledge
    - Enrich the Web with the knowledge of the best means and ends
    - Equip the Web with high discernment and judgement

Why the Wisdom Web?

- The content of the human:
  - Data→Information→Knowledge→Wisdom
- The content of the Web
  - Web data
  - Web information
  - Web knowledge
  - **Web wisdom?**



Fundamental Capabilities of the Wisdom Web

- Self-organizing servers
- Specialization
- Growth
- Autocatalysis
- PSML
- Semantics
- Metaknowledge
- Planning
- Personalization
- A sense of humor

Great Challenges for WI [Liu, 2003]

- Mobilizing distributed resources
  - Distributed Resource Optimization (DRO)
  - Web oriented computing paradigm
    - Large-scale
    - Pervasive and distributed
    - Dynamics and unreliable
    - Heterogeneous
    - …
  - …
- Discovering the best means and ends
- Enriching social interaction

Autonomy Oriented Computing (AOC)

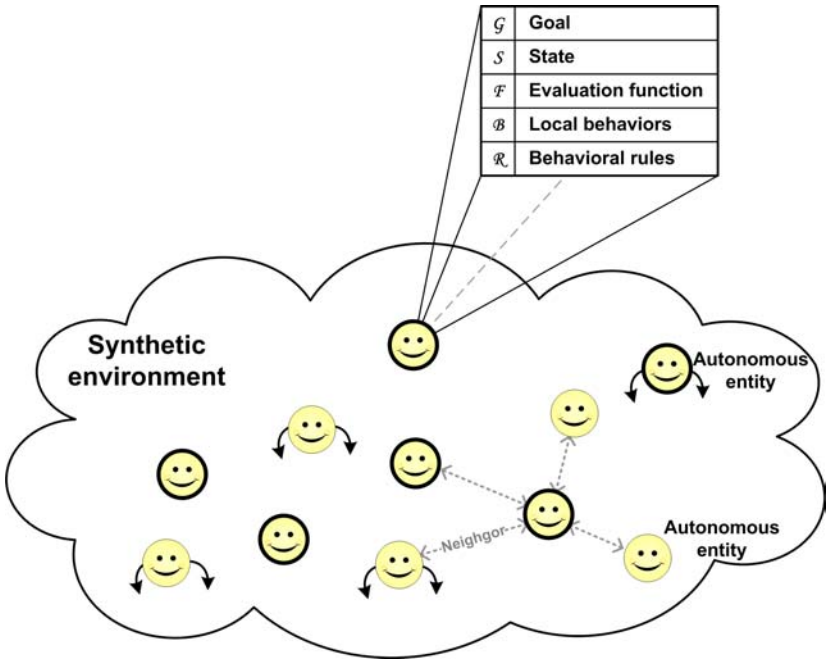
- A bottom-up approach for studying complex systems
- Proposal: by **Liu**, in 2001
- Purpose: develop computational algorithms or systems that employ autonomy as the core model of any complex system behavior
  - Complex systems modeling
  - Hard computational problem solving

Motivation and Inspiration

- Motivated by the need for solving large-scale, distributed problems and modeling complex systems
- Inspired by the autonomy and self-organization phenomena in nature

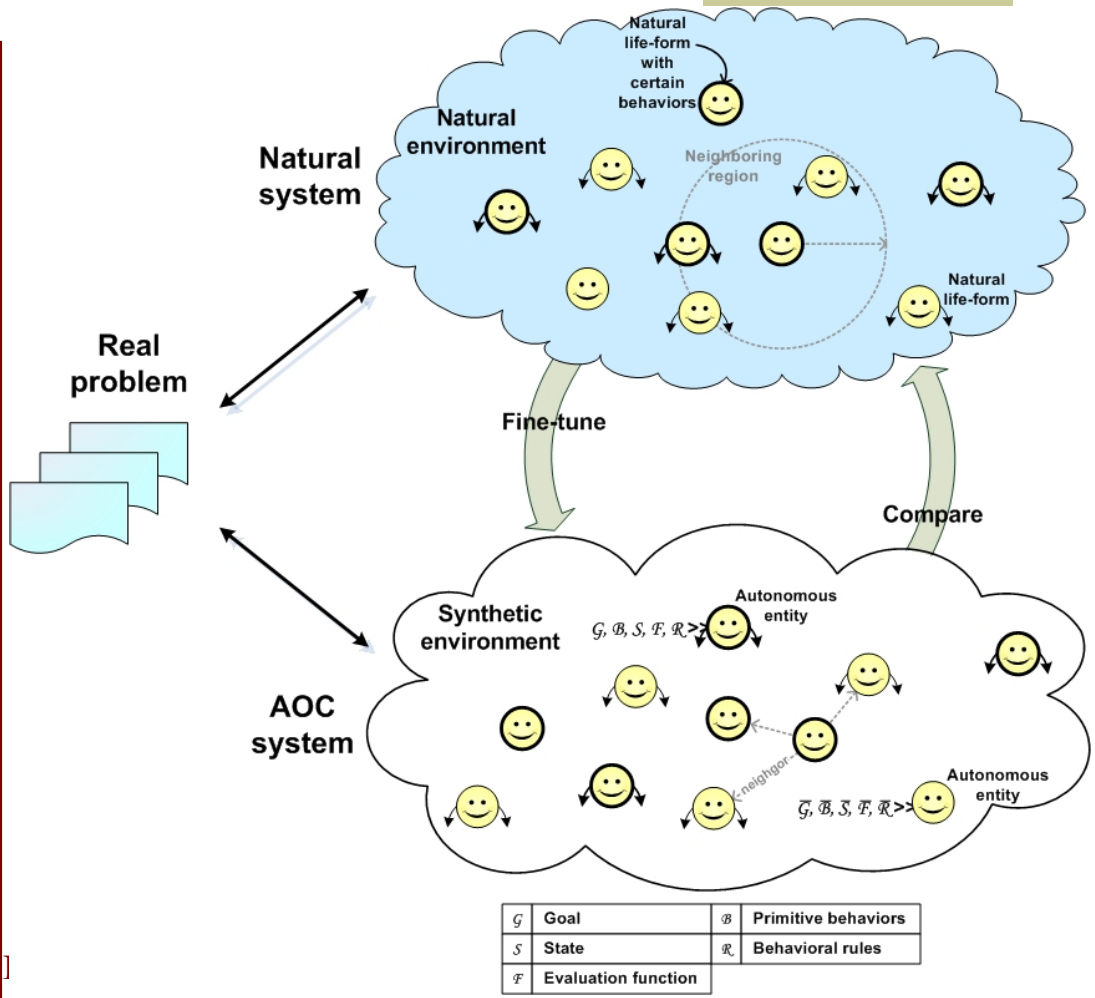
Components of An AOC System

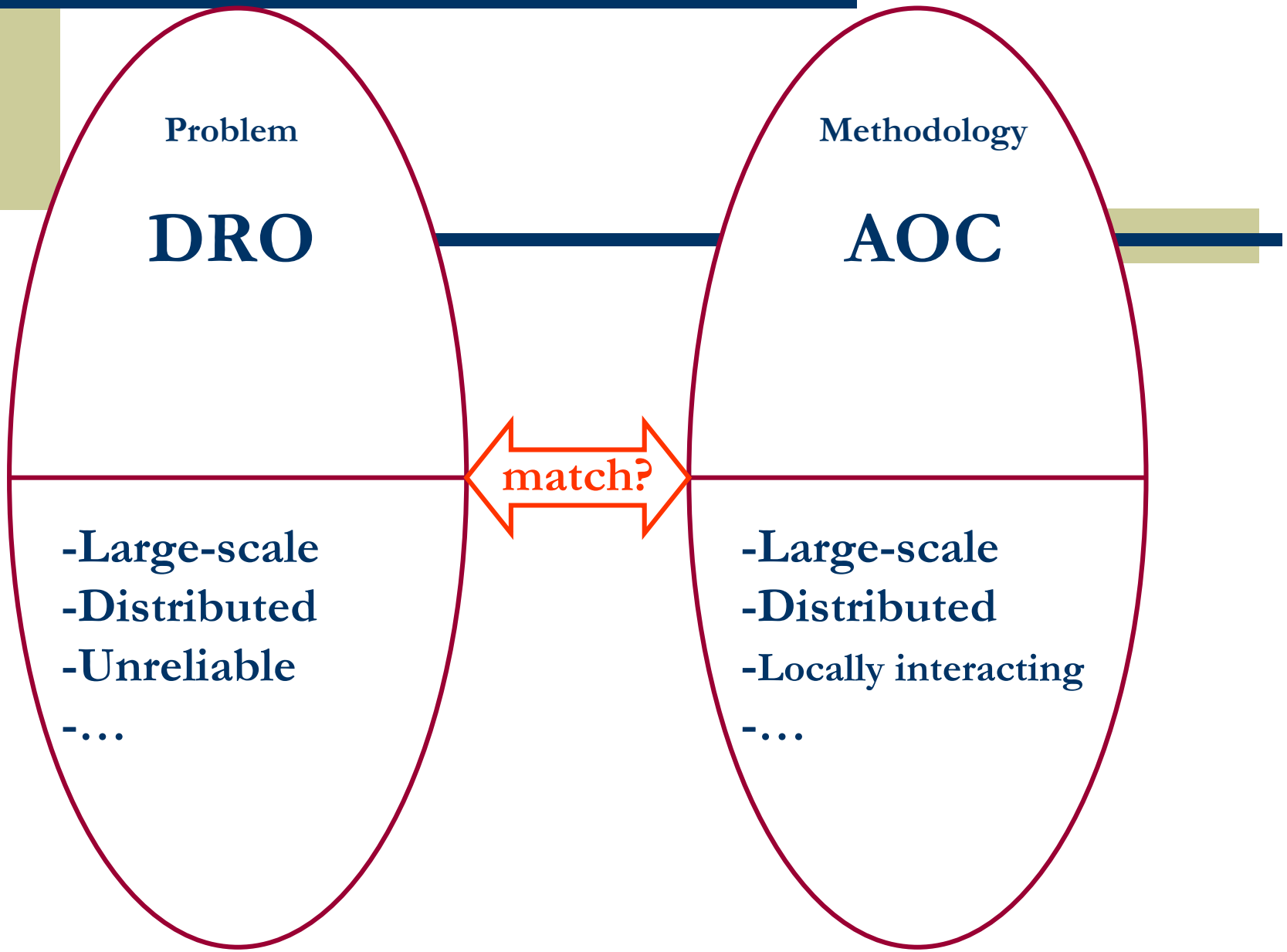
- Autonomous entities
- An environment
- Interactions
  - Between entities and the environment
  - Among entities
- System objective function
- Two important notions
  - Autonomy
  - Self-organization



Common Steps for Building an AOC System

1. Find out a certain natural phenomenon/system which is, to some extent, related to the problem or similar to the system at hand;
2. Observe the simple behaviors of the elemental entities in the natural system, particularly, their interactions with each other or the environment;
3. Build a certain mapping between a certain state of the natural system and a solution to the problem at hand;
4. Design synthetic entities and equip them with primitive behaviors which can be analogized to those of entities in the natural system;
5. Through analogy to the natural system, build a synthetic system with the above synthetic entities;
6. Run the synthetic system;
7. Observe the macroscopic behavior of the synthetic system as well as the microscopic behavior of its synthetic entities;
8. Compare the macroscopic and microscopic behaviors of the synthetic system to those of the natural system;
9. Modify steps 4-5 in view of step 8;
10. Repeat steps 4-9 until the synthetic system can (1) evolve to a state corresponding to a solution to the problem at hand, or (2) simulate the complex system to a satisfactory extent.




Problem
DRO
-Large-scale
-Distributed
-Unreliable
-…
match?
Methodology
AOC
-Large-scale
-Distributed
-Locally interacting
-…

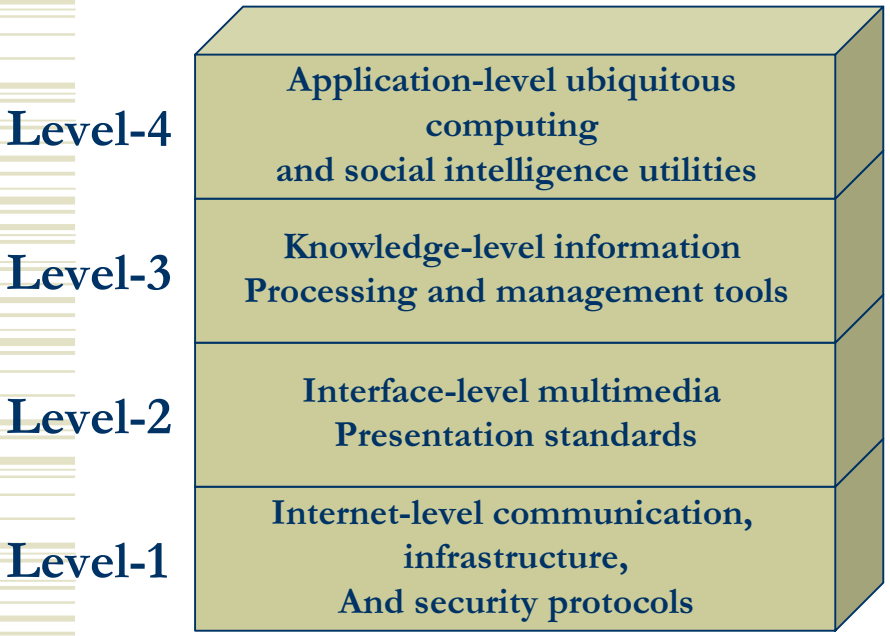
Research Problems

Research Scope and Problems

- Research scope: To design an AOC-based methodology for DRO on the Web
- Specific problems
  - **Q-1:** What resources to be optimized? What requirements?
  - **Q-2:** How to generalize specific DRO issues?
  - **Q-3:** How to provide an AOC-based computing paradigm for the generalized DRO?
  - **Q-4:** How to refine and validate the above paradigm according to the features of different DRO environments?
    - **Q-4-A:** Homogeneous DRO environments
    - **Q-4-B:** Heterogeneous DRO environments
- General assumptions:
  - Features considered: large-scale, distributed, unreliable, heterogeneous, and asynchronous
  - Not considered: security, privacy, interoperability, and transportation cost

Distributed Resource Optimization (DRO) Perspective on WI

Four Conceptual Levels of WI



- WI has been studied at four conceptual levels
  - Internet level
  - Interface level
  - Knowledge level
  - Application level

Q-1: DRO Perspective

- Generalized view of distributed resources
  - At different WI levels, resources may refer to different contents with different functions or utilities
  - A resource may be in a physical or a logical sense
- DRO at four WI levels
  - Internet level
    - E.g. CPU time & processing speed, network width, computers
    - E.g. How to distribute tasks to different computers such that the load is balanced?
  - Interface level
    - E.g. Portals
  - Knowledge level
    - E.g. distributed data/knowledge bases
  - Application level
    - E.g. various high-level Web services

Q-2: A Generalized DRO Scenario

- Assumptions
  - One service request needs only one resource to be served
  - No temporal, spacial, logical, or other dependency relationships exist among different service requests
  - Service requests follow some distribution collectively
  - Service request handling cost is dominant, as compared to network transportation cost
- Generalized scenario
  - Resources
    - A DRO environment is abstracted as a set of resource nodes and a set of links among them
    - Two directly linked resource nodes as *neighbors* to each other
    - Resource nodes are characterized with several parameters
    - Resource nodes can be homogeneous or heterogeneous
    - Resource nodes are unreliable
    - Resource nodes have capacities in terms of the service request load they can endure
    - Each resource node can provide one or more services

Q-2: A Generalized DRO Scenario
(Cont.)

- Service requests
  - Demands to resources are referred to as service requests
  - Service requests are submitted in a distributed fashion
  - No centralized mechanism is responsible for distributing service requests
  - Service requests are characterized by several parameters
  - Service requests may be homogeneous or heterogeneous
- Resource optimization
  - **How to distribute service requests among resource nodes such that their specific requirements are satisfied?**
  - **How to distribute service requests among resource nodes such that they are utilized in an optimized way, while keeping the load on resource nodes at an (approximately) balanced state?**
  - **Since resource nodes and service requests involved are large-scale and distributed, the resource optimization mechanism to be proposed for solving the above two problems should be scale-free and work in a distributed fashion**

Q-1: WI Requirements on DRO

- *Semantic* : Service requests should be semantically matched to the resources required
- *Correct* : The match between a request and a service should be correct, i.e., the resource node can definitely provide the service required by the service request
- *Distributed* : The mechanism for DRO should be distributed rather than centralized
- *Optimized* : The mechanism should try to achieve that the utilization of resources is in an optimal or sub-optimal fashion
- *Global* : The mechanism should concern DRO mainly at a global scale of the Web rather than at a local scale, because local optimization cannot guarantee global optimization

Q-1: WI Requirements on DRO
(Cont.)

- *Online* : The mechanism should make real-time decisions on the allocation of service requests to resource nodes
- *Robust* : The mechanism should be able to overcome various incidents
- *Adaptive* : The mechanism should adapt to real-time changes in the resource environment, particularly, in resources and service requests
- *Autonomic* : The mechanism should achieve the above requirements or implement the above functionality by itself

Q-3:
AOC Mechanism & Formulation for DRO

AOC-Based DRO Mechanism

- Agents carry service requests to search:
  - Idle resource nodes to form new agent teams
  - Existing agent teams to join
- Agents prefer to join agent teams with less load
- When queuing, agents can choose to
  - Remain at current agent teams, or
  - Leave current teams and wander to other resource nodes
- Agents have four primitive behaviors: *remain*, *wander*, *join*, and *leave*
- Agents must be served by a certain resource node to have its service request handled
- Agents automatically disappear after after being served
- Agents have only local information about its service request, current resource node and agent team, as well as the resource nodes in its neighboring region
- Agents indirectly interact via the environment

Resource Environment

- Environment *E*: a graph $\langle V, L \rangle$, where $V = \langle rn_1, \cdots, rn_i, \cdots, rn_N \rangle$ is a set of resource nodes, and $L = \langle l_1, \cdots, l_i, \cdots, l_K \rangle$ is a set of links among resource nodes
- Resource node *rn*: a 7-tuple, $rn = \langle fl, si, ps, qat, qts, wat, nl \rangle$, where
  - *si*: service vector
  - *ps*: processing speed vector corresponding to *si*
  - *qts*: the size of agent team at node *rn*, i.e., *qat*

Agents

- State: a 12-tuple, $S=\langle wq, vr, pos, hn, rq, rs, dl, nab, tlw, rt, tlh, ct\rangle$ where
  - *wq* : wandering (1) or queuing (0)
  - *vr* : vision range
  - *pos* : position
  - *rt* : response time
- Neighboring region: $NR=\langle V', L'\rangle$, where $V'\in V$ and $L'\in L$. Particularly, (1) $\forall rn_i\in V$, if $hpc(rn_i,a.pos)\leq a.vr$, then $rn_i\in V'$; (2) $\forall\langle rn_i,rn_j\rangle\in L$, if $hpc(rn_i,a.pos)\leq a.vr$ and $hpc(rn_j,a.pos)\leq a.vr$, then $\langle rn_i,rn_j\rangle\in L'$
- Neighbors: $a.NR=\langle V', L'\rangle$ and $b.pos=rn_i$ and $rn_i\in V'$, *b* is a neighbor of agent *a*

Evaluation Functions and Goal

- Evaluation functions: $F=\{f_s, f_l\}$
  - f_s : returning the state of being wandering or queuing of agent a
  - f_l :evaluating the load of a resource node rn in the neighboring region of agent a

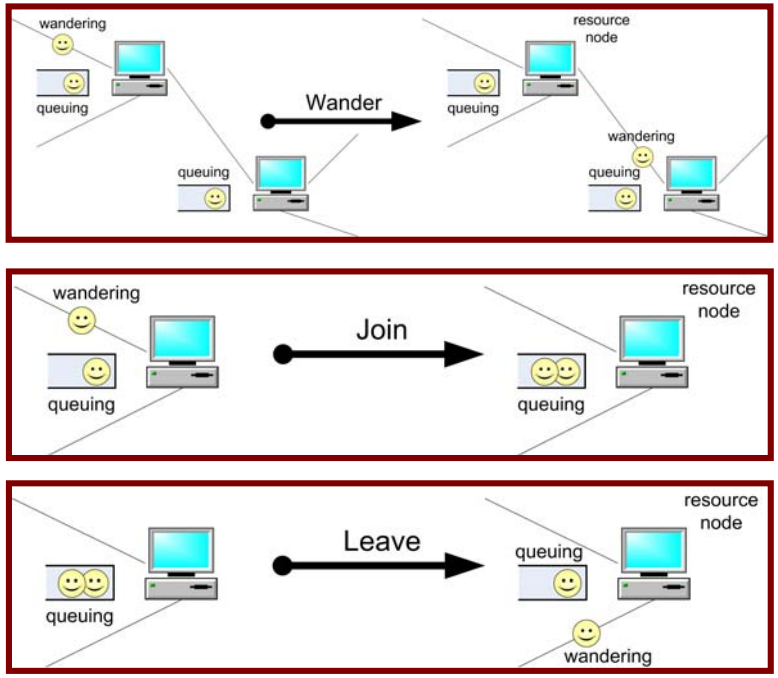
$$f_l(rn) = f_l(rn.qat) = \sum_{b \in rn.qat} b.rq, \quad \text{or}$$

$$f_l(rn) = f_l(rn.qat, rn.ps) = \sum_{b \in rn.qat} \frac{b.rq}{rn.ps_{(b.rs)}}$$

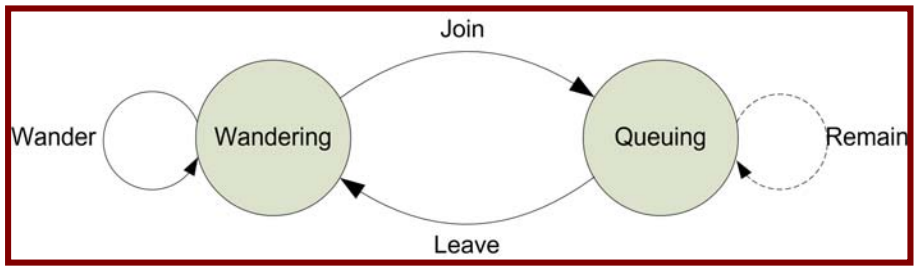
- Goal: $g : a.pos = u, \ \text{where } u = \arg\min_{rn \in V}(f_l(rn))$

Primitive Behaviors

- Primitive behaviors: *B={remain, wander, join, leave}*

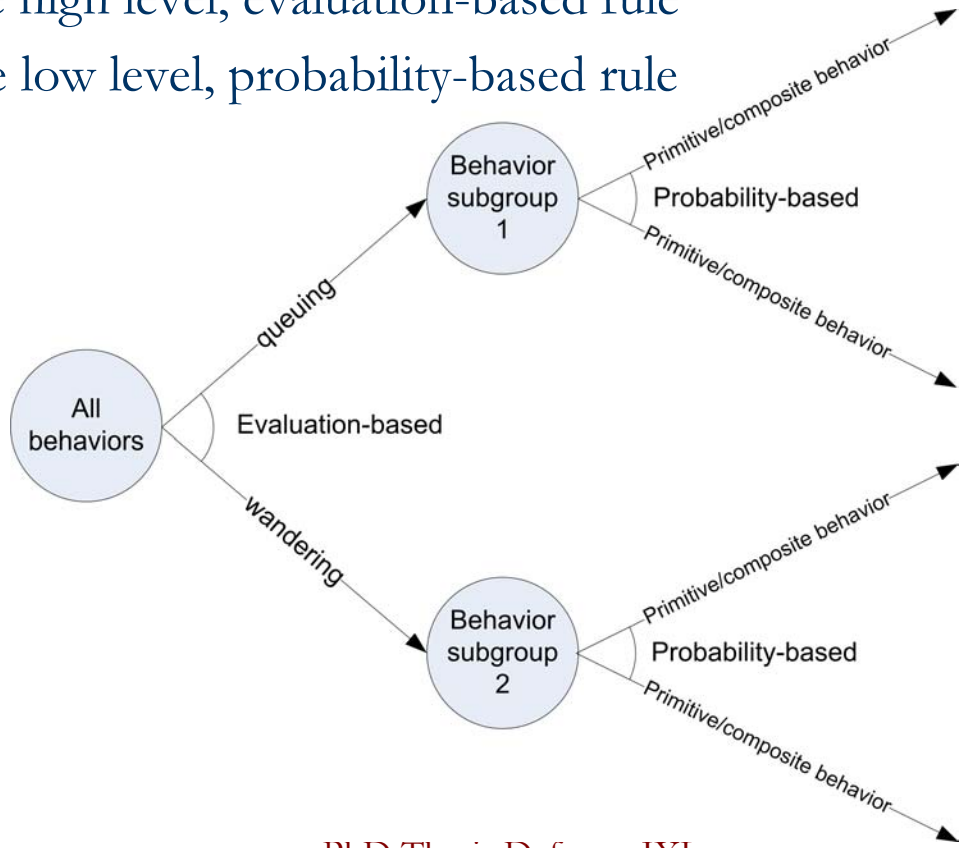


State Transition

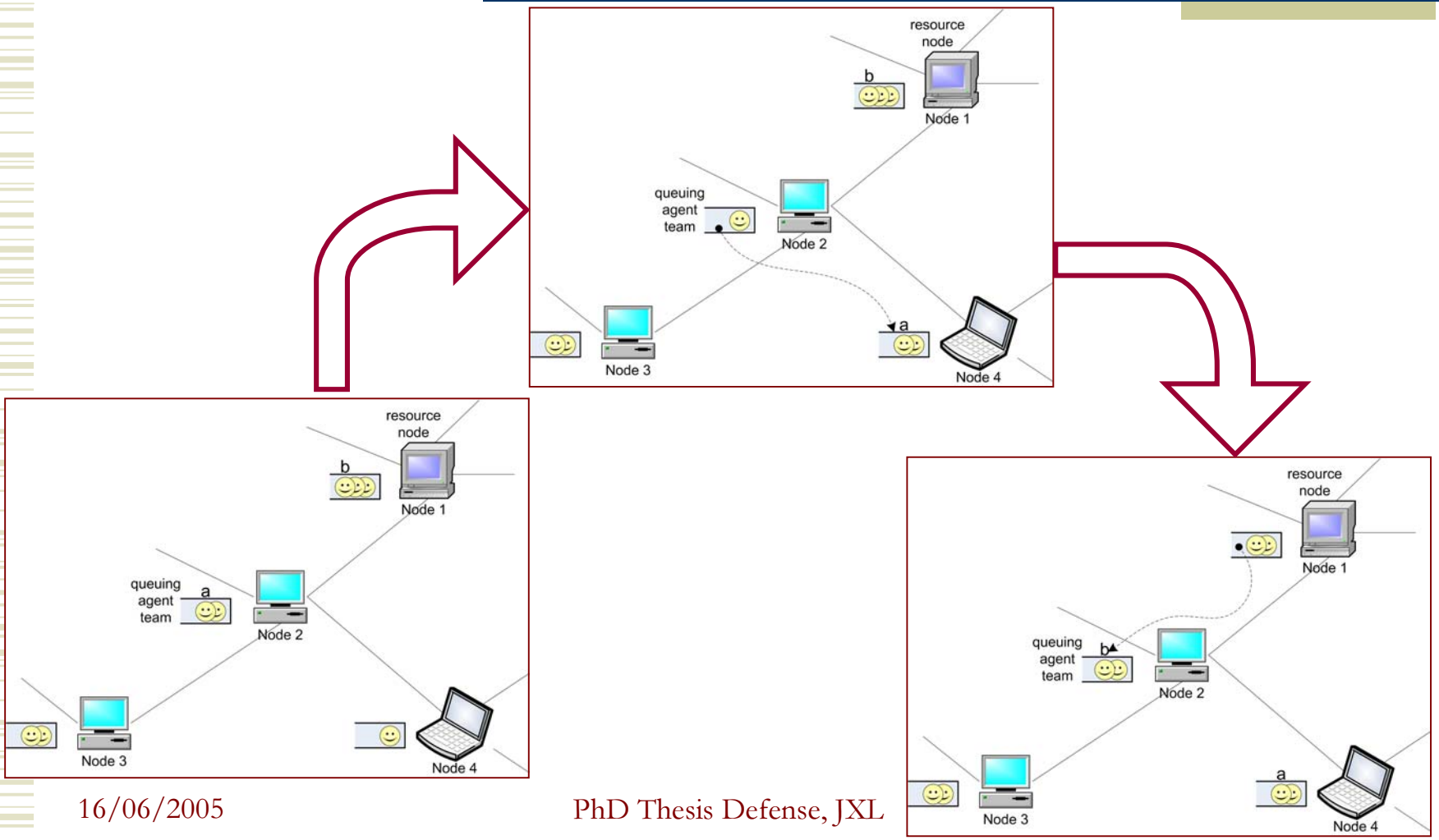


Behavioral Rules

- Behavioral rules: $R=\{r_h, r_l\}$
  - r_h : the high level, evaluation-based rule
  - r_l : the low level, probability-based rule



Indirect Interactions among Agents



System Objective Function

$$\Phi(\mathbf{V},\mathbf{A}) = std_{rl} = \sqrt{\frac{\sum_{j=1}^{N}(rn_j.rl - \frac{\sum_{i=1}^{N} rn_i.rl}{N})^2}{N}}$$

- $\boldsymbol{V}=\{rn_1,L,rn_i,L,rn_N\}$: the set of resource nodes
- $\boldsymbol{A}=\{rn_1.qat, L,rn_i.qat,L,rn_N.qat\}$: the set of queuing agent teams at resource nodes in $\boldsymbol{V}$
- $rn_i.rl = \sum_{a \in rn_i.qat} \frac{a.rq}{rn_i.ps_{(a.rs)}}$

Q-4-A:
DRO in Homogeneous Environments

Homogeneous Resource Environments Considered

- Resource nodes are homogeneous
  - Providing the same service
  - Having the same processing speed
  - e.g., DAS
- Service requests are homogeneous
  - Requiring the same services
  - Having the same size
  - e.g., DLT

Refined DRO Mechanism

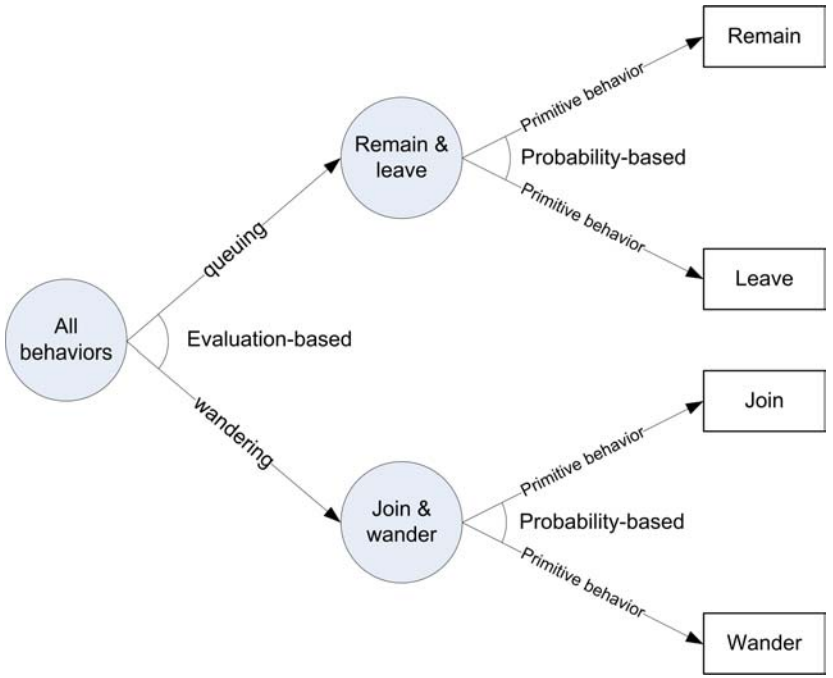
- The size of an agent team = Service request load
- Agents' decision-making based on
  - Probabilities
  - The size of agent teams encountered
- Agents' preference: relatively small agent teams
- Resource nodes' capability : A maximum size for agent teams
- At each step, how many wandering agents join agent teams of a certain size depends on:
  - The total number of currently wandering agents
  - The numbers of currently existing agent teams of various sizes
- At each step, how many queuing agents leave teams of a certain size depends on:
  - The total number of currently existing teams of this size
- In the above sense, the proposed mechanism is adaptive

Refined AOC Formulation

- Homogeneous environments
  - all resource nodes, *rn*, provide the same service *ri*, with the same processing speed *ps* and the same service request load capacity *cp*
  - The load of resource node *rn*: $rn.l = rn.qts$
  - The maximum team size *m*, indicating the capacity of resource nodes, i.e., $rn.cp = m$. It should guarantee $rn.l = rn.qts \leq m$
  - ϖ and ϑ denote the average service request load among resource nodes and its standard deviation

Agents

- State:
  - Two vectors: $\langle \zeta_0(t), \cdots, \zeta_s(t), \cdots, \zeta_{m-1}(t) \rangle$ and $\langle \iota_2(t), \cdots, \iota_s(t), \cdots, \iota_m(t) \rangle$
    - $\zeta_s(t)$: probability for joining teams of size s
    - $\iota_s(t)$: probability for leaving teams of size s
    - $\zeta_s(t)$ and $\iota_s(t)$ are fixed over time
  - Vision range $a.vr=1$
- Evaluation function: $f_l(rn)=rn.qts$
- Behavioral rule



- System objective function: $\Phi(\mathbf{V}, \mathbf{A}) = \sqrt{\frac{\sum_{j=1}^{N}(rn_j.l - \frac{\sum_{i=1}^{N} rn_i.l}{N})^2}{N}}$

Performance Studies

- Instantaneous DRO Scenario
  - A small time interval ➔ no new service request + no handled service request
  - **I-1.** Can the mechanism achieve a steady state where load is balanced?
  - **I-2.** Given different resource environment, can the mechanism adapt it and achieve the (approximately) optimal resource utilization?
  - **I-3.** Is the mechanism robust to tolerate the dynamic changes in the environment and adapt their outcome?
- Ongoing DRO Scenario
  - A long time interval➔ new service requests + handled service requests
  - **I-4.** How does the arrival speed of service requests affect the performance? How to determine an appropriate arrival speed?
  - **I-5.** Is the mechanism robust to tolerate the dynamic changes in the environment and adapt their outcome?

AOC-based Instantaneous DRO Model

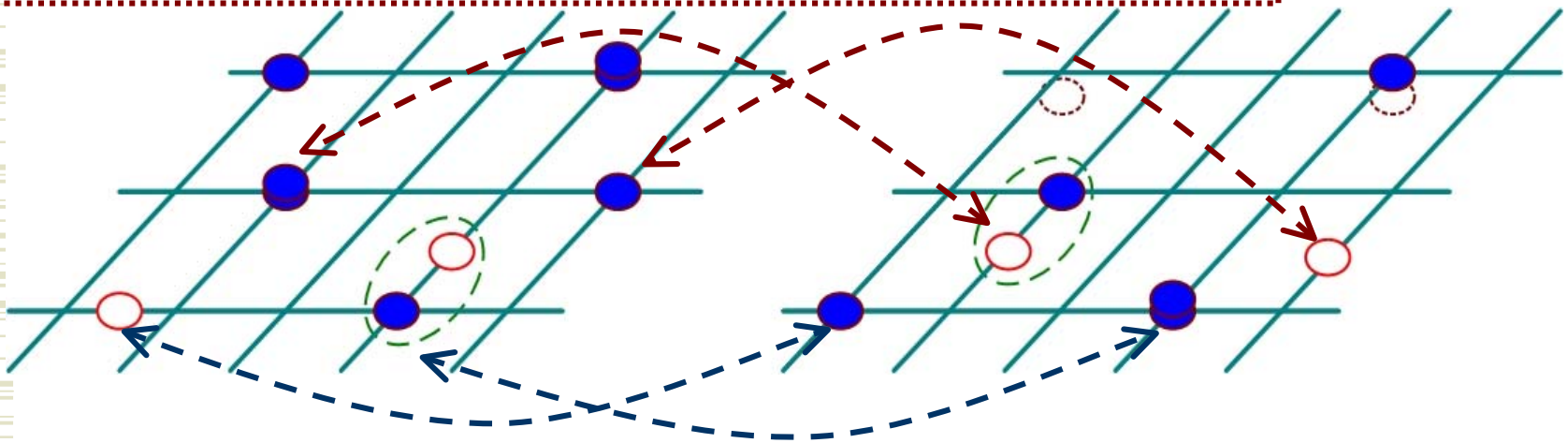
(Case: m=3)

The quantitative changes of:

Agents teams of size one:

$$\frac{dq_1(t)}{dt} = j_0 w(t) - j_1 w(t) + l_2 q_2(t)$$

- Wandering agents join idle nodes or existing agent teams
- Queuing agents leave existing agent teams



The quantitative changes of:

Agents teams of size two:

$$\frac{dq_2(t)}{dt} = j_1 w(t) - l_2 q_2(t)$$

Wandering agents:

$$\frac{dw(t)}{dt} = l_2 q_2(t) - \sum_{s=0}^{1} j_s w(t)$$

Idle resource nodes:

$$\frac{dq_0(t)}{dt} = -j_0 w(t)$$

A General Model

$$\frac{dq_1(t)}{dt} = j_0 w(t) - j_1 w(t) + l_2 q_2(t)$$

$$\frac{dq_s(t)}{dt} = j_{s-1} w(t) - j_s w(t) + l_{s+1} q_{s+1}(t) - l_s q_s(t)$$

$$\frac{dq_m(t)}{dt} = j_{m-1} w(t) - l_m q_m(t)$$

$$\frac{dq_0(t)}{dt} = -j_0 w(t)$$

$$\frac{dw(t)}{dt} = \sum_{s=2}^{m} l_s q_s(t) - \sum_{s=0}^{m-1} j_s w(t)$$

$$j_s(t) = \begin{cases} \frac{q_s(t)}{w(t)}, & \text{if } \bar{j}_s(t) - \frac{q_s(t)}{w(t)} \geq 0, \\ \frac{q_s(t)}{w(t)}, & \text{if } \frac{\Phi(t)}{\Psi(t)} \geq 1, \\ \bar{j}_s(t) + \frac{\Phi(t)}{\Psi(t)}(\frac{q_s(t)}{w(t)} - \bar{j}_s(t)), & \text{otherwise,} \end{cases}$$

where

$$\bar{j}_s(t) = \frac{j_s^p \cdot j_s^d(t)}{\sum_{i=0}^{m-1}(j_i^p \cdot j_i^d(t))} o$$

$$\Phi(t) = \sum_{i=0}^{m-1} sgn(\bar{j}_s(t) - \frac{q_s(t)}{w(t)})(\bar{j}_s(t) - \frac{q_s(t)}{w(t)}),$$

$$\Psi(t) = \sum_{i=0}^{m-1} sgn(\frac{q_s(t)}{w(t)} - \bar{j}_s(t))(\frac{q_s(t)}{w(t)} - \bar{j}_s(t)),$$

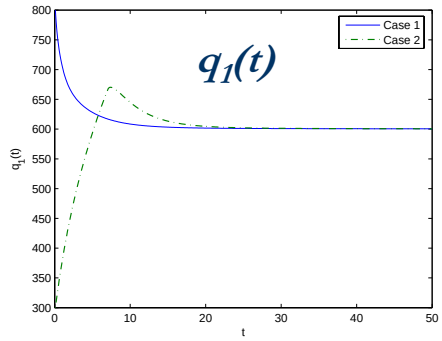
$$sgn(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{otherwise.} \end{cases}$$

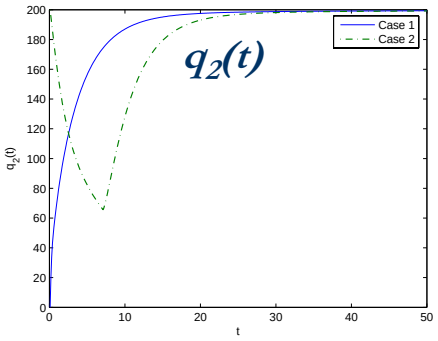
$$l_s(t) = \begin{cases} \bar{l}_s(t), & \text{if } N = 0, \\ \frac{\bar{l}_s(t)}{N+1}, & \text{otherwise,} \end{cases}$$

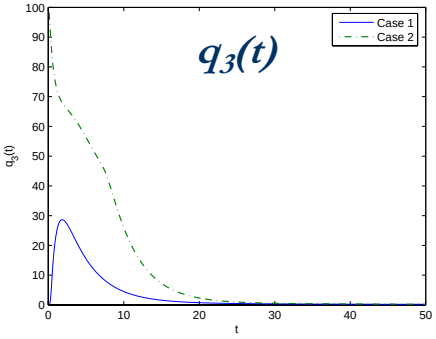
where

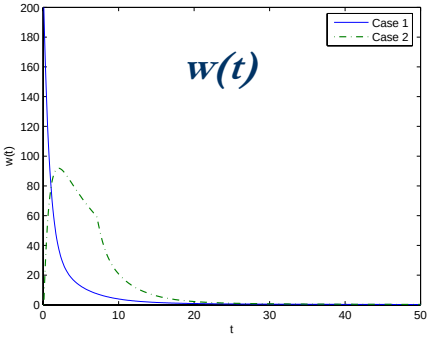
$$\bar{l}_s(t) = \frac{l_s^p \cdot l_s^d(t)}{\sum_{i=2}^{m}(l_i^p \cdot l_i^d(t))},$$

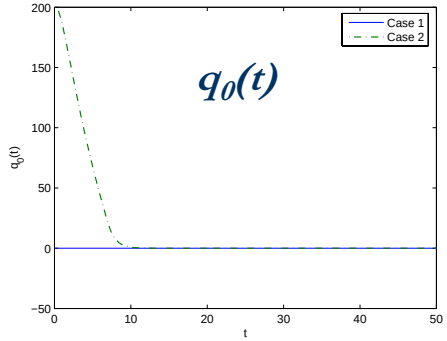
and N is the consecutive times when $\forall i \in \{1, 2, \cdots, N\}$, $j_{s-1}(t-i) > \bar{j}_{s-1}(t-i)$ and $j_{s-1}(t-N-1) \leq \bar{j}_{s-1}(t-N-1)$.

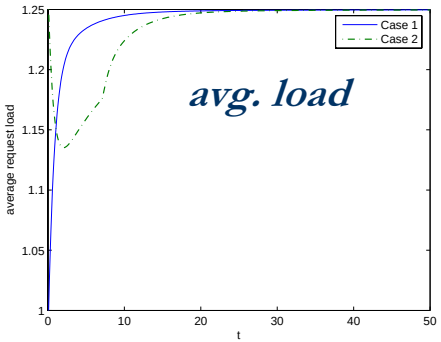


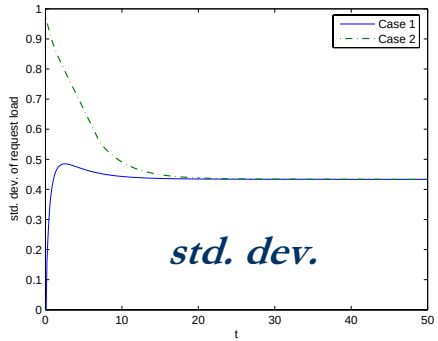






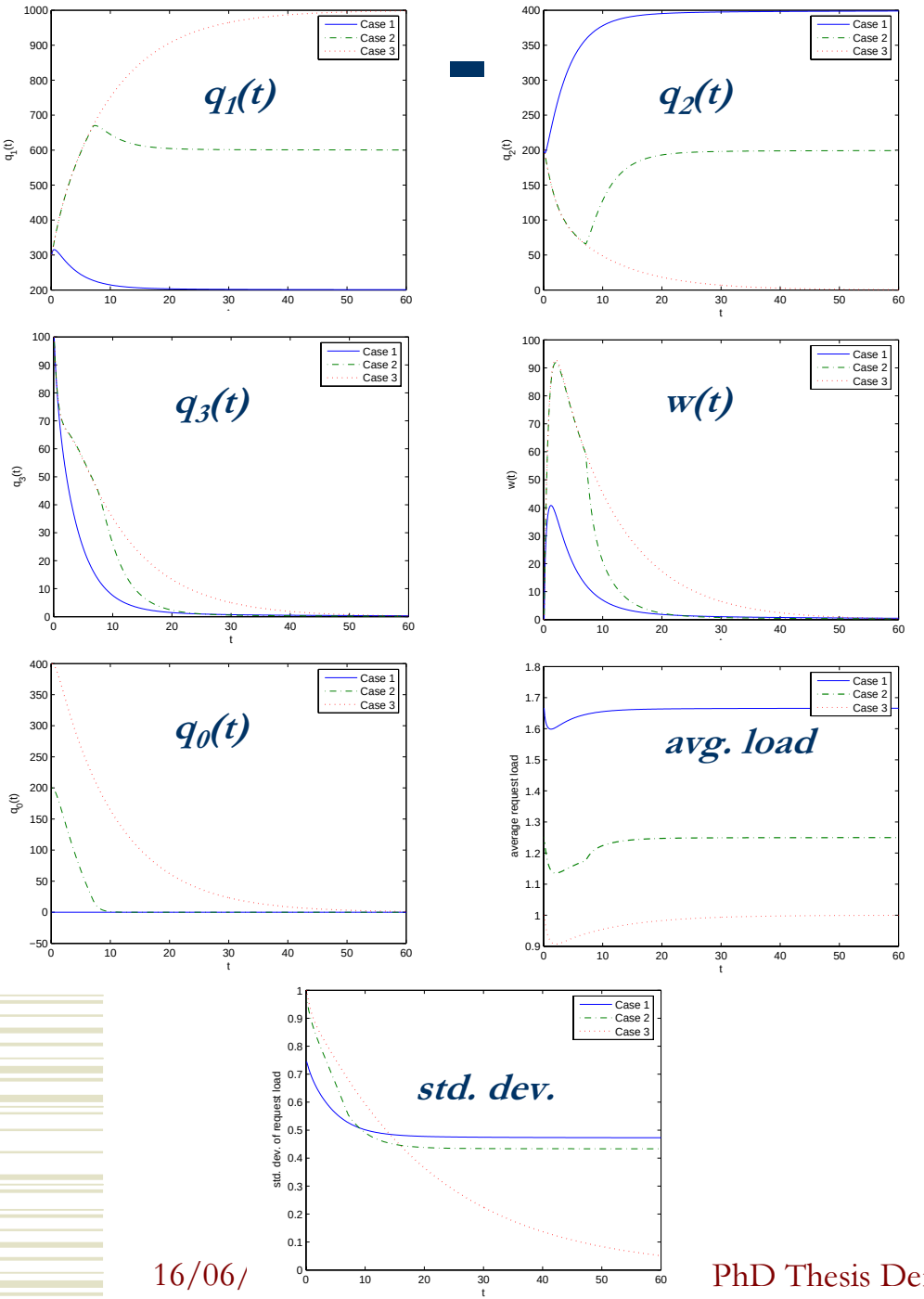






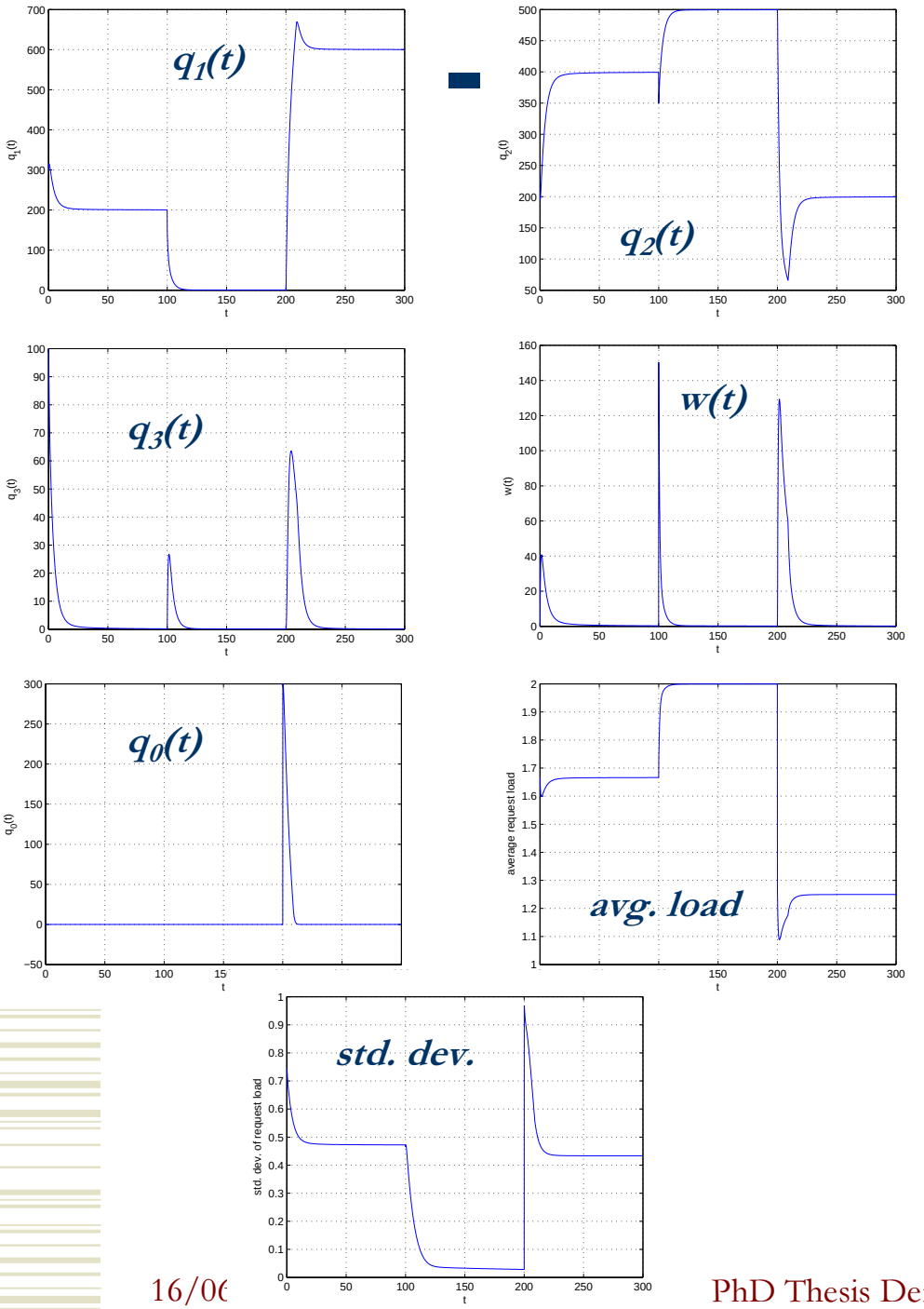
I-1. Study on: Global stability

- **Setting**
  - Case I: $q_0(0)=0$, $q_1(0)=800$, $q_2(0)=0$, $q_3(0)=0$, $w(0)=200$
  - Case II: $q_0(0)=200$, $q_1(0)=300$, $q_2(0)=200$, $q_3(0)=100$, $w(0)=0$
- **Observations**
  - Any initial agent distribution →
    - a steady state, a load-balanced state, an optimal resource utilization
    - all characterizing parameters, nonnegative
  - Different agent distributions → the same steady state → globally stable



I-2. Study on: Adaptation

- **Setting:** $m=3$, $S(0)=1000$, $q_0(0)=300$, $q_1(0)=300$, $q_2(0)=200$, $q_3(0)=100$, $w(0)=0$
  - Case I: $q_0(0)=0$, $Q(0)=600$
  - Case II: $q_0(0)=200$, $Q(0)=800$
  - Case III: $q_0(0)=400$, $Q(0)=1000$
- **Observation**
  - Given different numbers of service requests (i.e., different resource environments), the proposed DRO mechanism can finally achieve states where:
    - The load (i.e., the number of agents) among different resource nodes is perfectly balanced
    - The resource are optimally utilized



I-3. Study on: Robustness

- **Setting:**
  - $m=3$, $S(0)=1000$, $q_1(0)=300$, $q_2(0)=200$, $q_3(0)=100$, $w(0)=0$, $Q(0)=600$, $Q(100)=500$, *and* $Q(200)=800$
  - At time $t=100$: *100* nodes fail
  - At time $t=200$: *200* new nodes added
- **Observations**
  - Successfully enduing relatively large-scale resource failures
    - Converging to steady states
  - Quickly responding to a drastic increase in the availability of resource nodes so as to re-balance the load
  - In general,
    - Robust to tolerate dynamic changes
    - Promptly adapting the results of dynamic changes

AOC-based Ongoing DRO Model

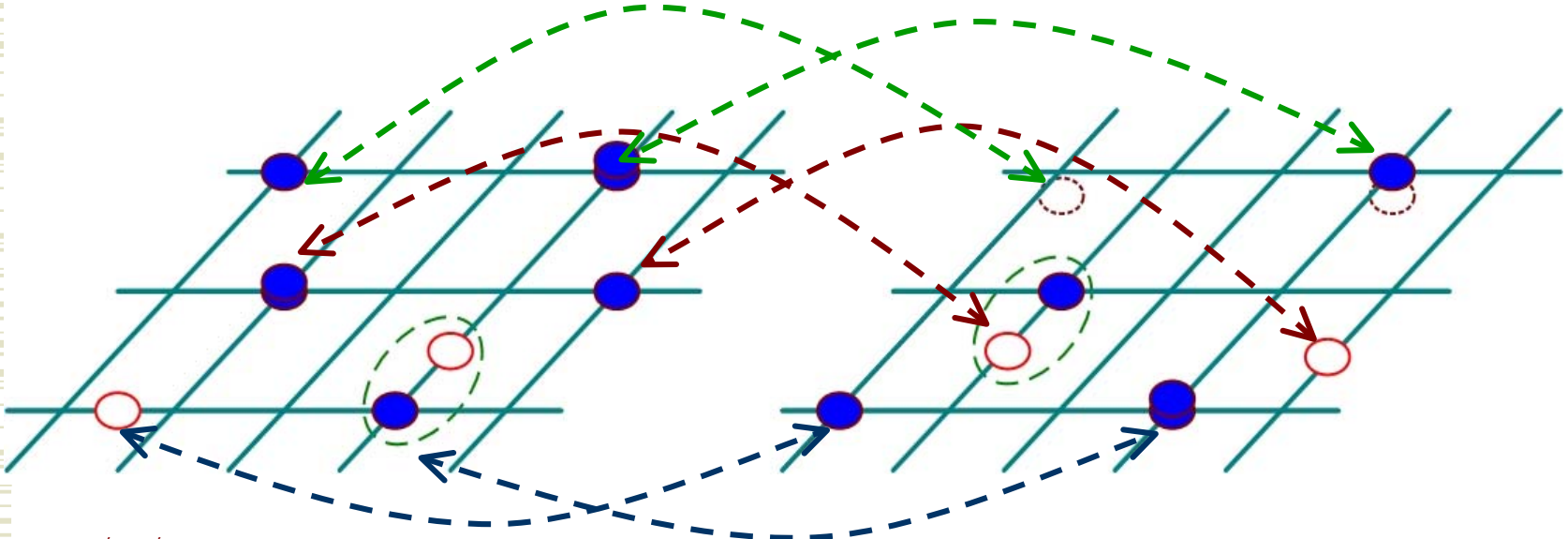
(Case: m=3)

The quantitative changes of:

Agents teams of size one:

$$\frac{dq_1(t)}{dt} = j_0 w(t) - j_1 w(t) + l_2 q_2(t) + f_2 q_2(t) - f_1 q_1(t)$$

- Wandering agents join idle nodes or existing agent teams
- Queuing agents leave existing agent teams
- Old service requests are finished after they are served a unit of service time



The quantitative changes of:

Agents teams of size two:

$$\frac{dq_2(t)}{dt} = j_1 w(t) - l_2 q_2(t) - f_2 q_2(t)$$

Wandering agents:

$$\frac{dw(t)}{dt} = l_2 q_2(t) - \sum_{s=0}^{1} j_s w(t) + g(t)$$

- *g(t)* : newly generated agents for new tasks

Idle resource nodes:

$$\frac{dq_0(t)}{dt} = -j_0 w(t) + f_1 q_1(t)$$

A General Model

$$\frac{dq_1(t)}{dt} = j_0 w(t) - j_1 w(t) + l_2 q_2(t) + f_2 q_2(t) - f_1 q_1(t)$$

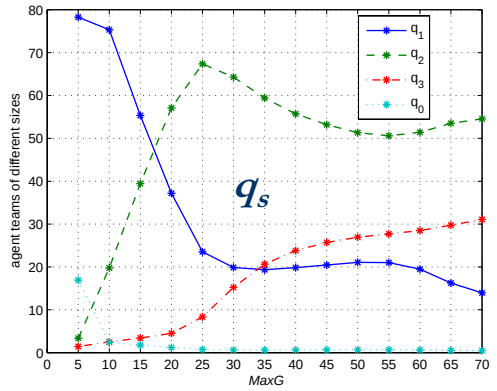
$$\frac{dq_s(t)}{dt} = j_{s-1} w(t) - j_s w(t) + l_{s+1} q_{s+1}(t) - l_s q_s(t) + f_{s+1} q_{s+1}(t) - f_s q_s(t)$$

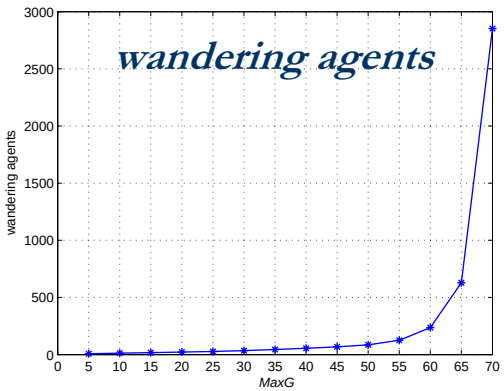
$$\frac{dq_m(t)}{dt} = j_{m-1} w(t) - l_m q_m(t) - f_m q_m(t)$$

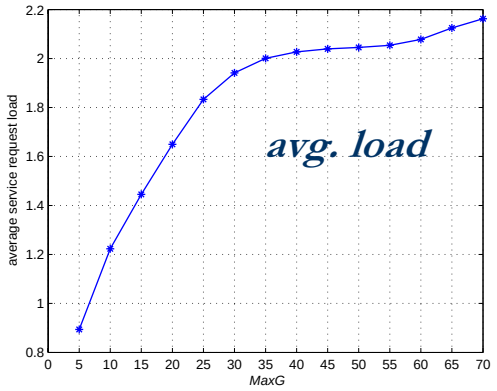
$$\frac{dq_0(t)}{dt} = -j_0 w(t) + f_1 q_1(t)$$

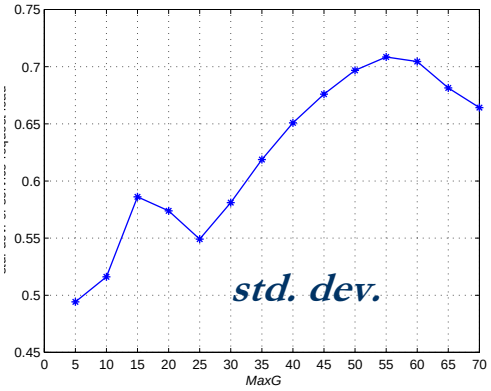
$$\frac{dw(t)}{dt} = \sum_{s=2}^{m} l_s q_s(t) - \sum_{s=0}^{m-1} j_s w(t) + g(t)$$

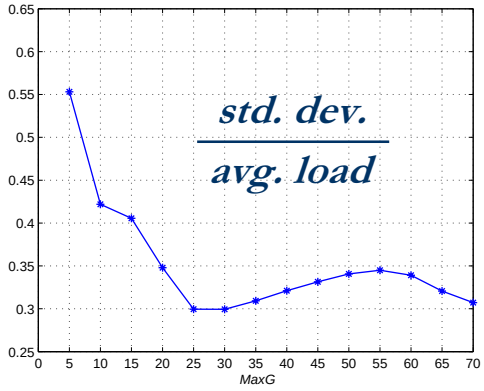
I-4. Study on: The effects of arrival speeds of service requests







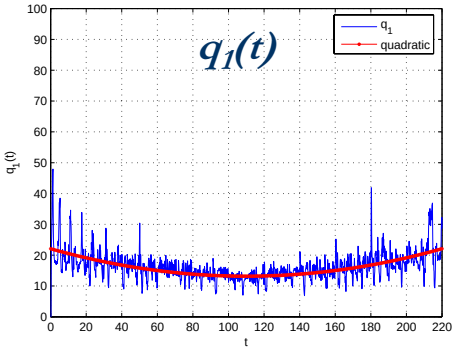


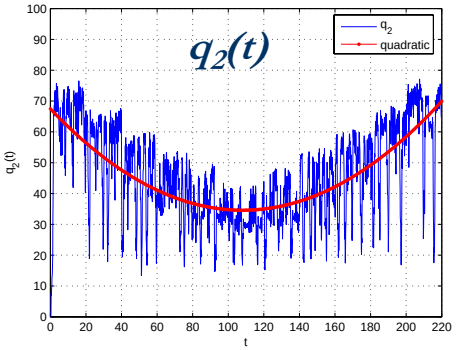


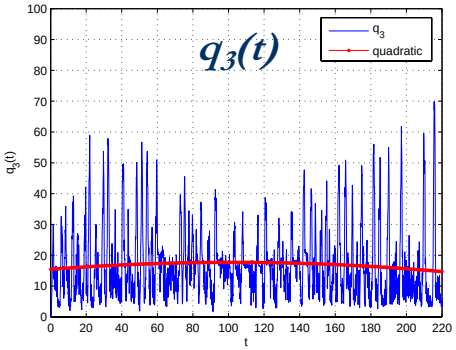
- **Setting:**
  - $g(t)=random([1,MaxG])$
  - $m=3$, $\lambda=10$, $Q(0)=100$, $S(0)=0$, $q_1(0)=0$, $q_2(0)=0$, $q_3(0)=0$, $w(0)=0$
- **Observations**
  - The arrival speed greatly affects the performance of the proposed mechanism. Fixing the service time of service requests,
    - A small arrival speed → less-loaded
    - A large arrival speed → over-loaded
  - Fixing service time λ, an appropriate arrival speed should be set according:

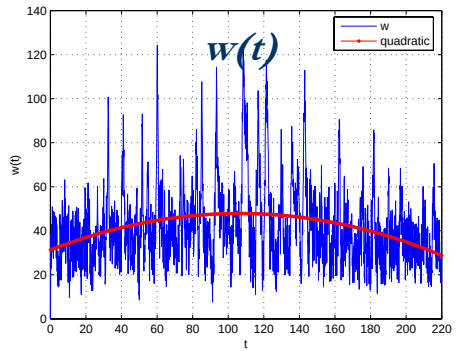
    $$MaxG \approx (m \cdot Q)/\lambda$$
  - The service requests arrived during a unit of service time (i.e., $MaxG \cdot \lambda$) should approximate the capacity of the whole resource environment (i.e., $m \cdot Q$):

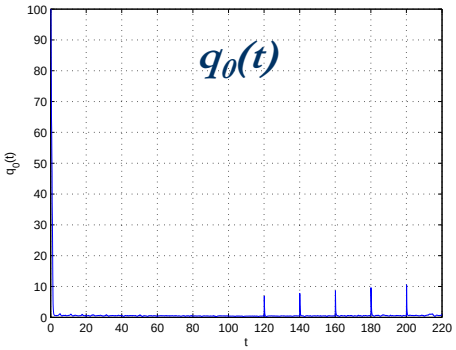
    $$MaxG \cdot \lambda \approx m \cdot Q$$

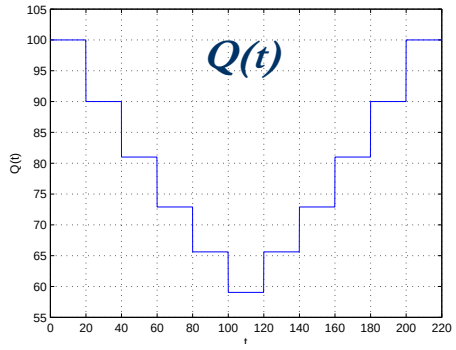


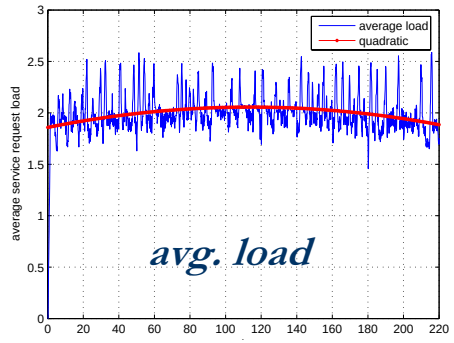


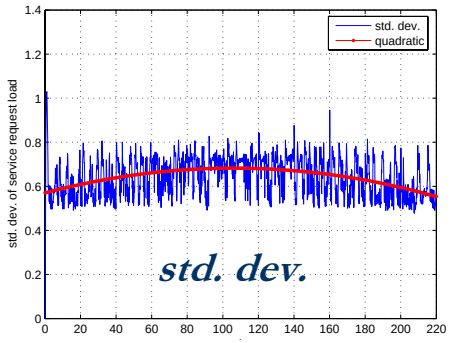








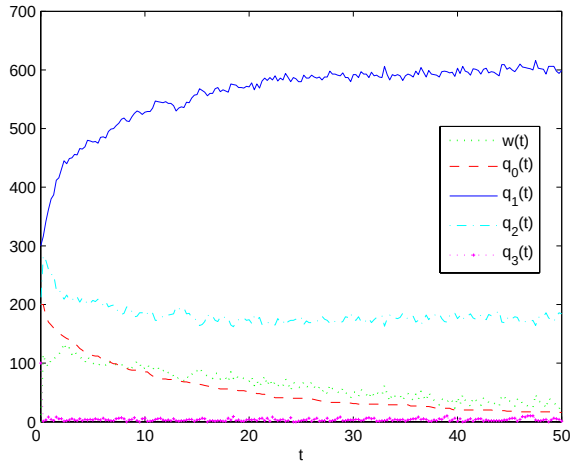


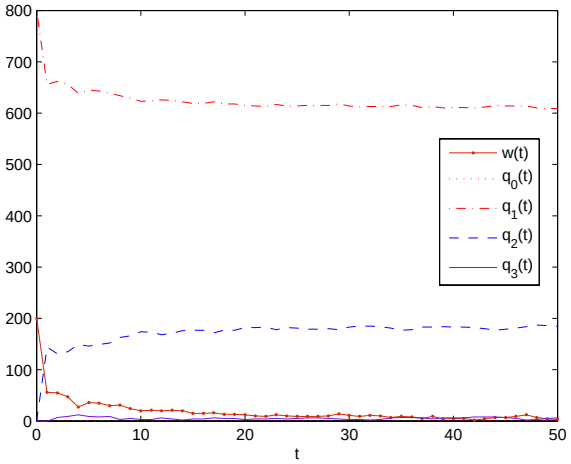


I-5. **Study on:** Robustness and adaptation

- **Setting:**
  - *MaxG=30*
  - *m=3*, λ *=10*, *Q(0)=100*, *S(0)=0*, $q_1(0)=0$, $q_2(0)=0$, $q_3(0)=0$, *w(0)=0*
  - At the first *100* steps : *0.1* percent of resource nodes failure per *20* steps
    - *0.1* percent of $q_0(t)$, $q_1(t)$, $q_2(t)$, $q_3(t)$ failure, respectively
  - At the later *100* steps : *0.1* percent of resource nodes recovered and are added to $q_0(t)$, per *20* steps
  - If a resource node with an agent team fails, queuing agents at this node become wandering agents
- **Observation**
  - The avg. service requst load and its std. dev. : no great changes
    - Successfully enduring resource failures
    - Quickly responding to increases in the availability of resource nodes
  - In general, the proposed mechanism is robust
    - It can tolerate failures and recovery of resource nodes without being greatly affected its performance

Experimental Validation

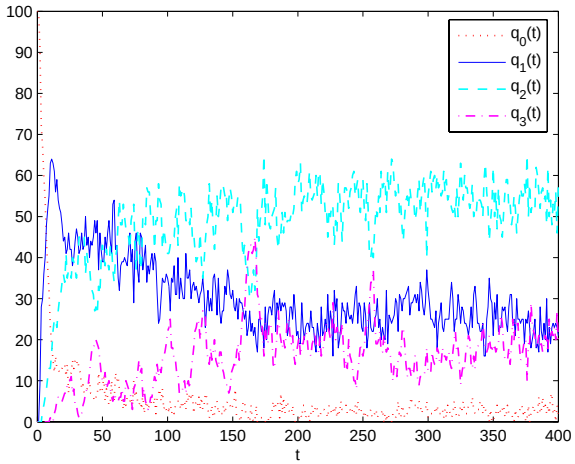


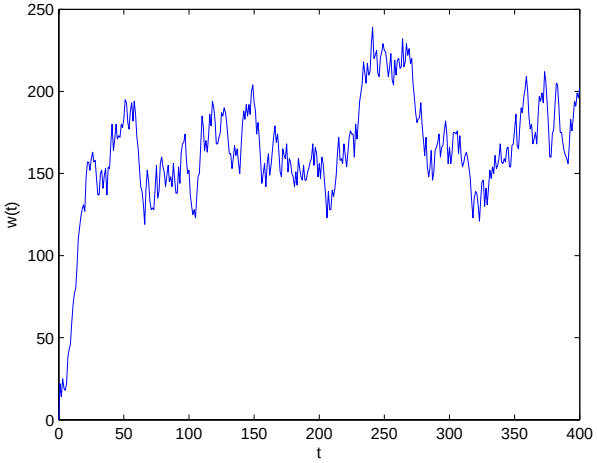


		$w(t)$	$q_0(t)$	$q_1(t)$	$q_2(t)$	$q_3(t)$	Avg. load	Std. dev.
The left figure	N	0	0	600	200	0	1.25	0.44
	E	1	0	604	193	3	1.249	0.441
The right figure	N	0	0	600	200	0	1.25	0.44
	E	30	16	599	184	1	1.212	0.458

Note: N – Numerical simulation; E – Experimental validation.

Experimental Validation (Cont.)





	$w(t)$	$q_0(t)$	$q_1(t)$	$q_2(t)$	$q_3(t)$	Avg. load	Std. dev.
Numerical Simulation	0	20	64	15	35	1.90	0.57
Experimental validation	3	25	53	19	163	1.88	0.73

Summary

- Instantaneous DRO Scenario
  - Given any initial agent distribution → a steady, load-balanced state
  - Different initial agent distributions → the same steady state → the proposed mechanism is globally stable
  - The proposed mechanism can tolerate large-scale failures and recovery of resource nodes → it is robust to endure dynamic changes occurred, adapt them, and finally reach a new steady state
- Ongoing DRO scenario
  - The arrival speed of service requests greatly affects the performance of the proposed mechanism
  - An appropriate arrival speed of service requests should be set according to:

    $$MaxG \approx \{m \cdot Q\} / \lambda$$

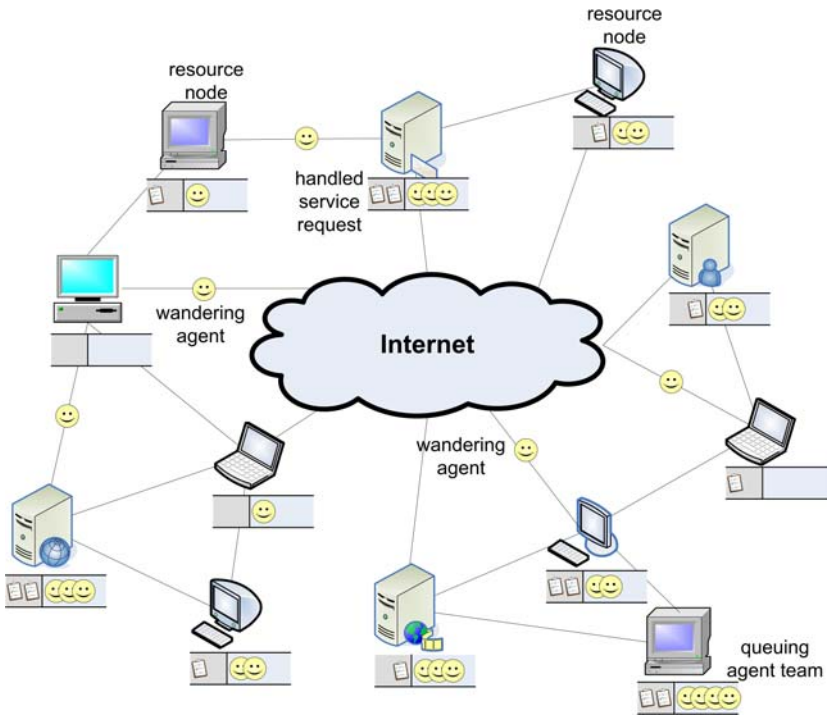
  - The proposed AOC-based DRO mechanism is robust and adaptive to tolerate failures and recovery of resource nodes without being greatly affected its performance

Q-4-B:
DRO in Heterogeneous Environments

Characterization of Heterogeneous Environments

- Heterogeneous resources
  - Different services
  - Different processing speed
- Heterogeneous service requests
  - Requiring different services
  - Different sizes
- Topology of resource networks
  - scale-free with a power of *3*
- Service request characterization
  - *interarrival times*, *sizes*, and *service times* ~ exponential distribution
  - λ_{iat} : exponential distribution of interarrival times
  - λ_{ts} : exponential distribution of sizes
- Failures and recovery of resource nodes
  - Exponential distributions
  - λ_{fti} : exponential distribution of failures
  - λ_{rti} : exponential distribution of recovery

Refined AOC Mechanism



- Agents are employed to carry service requests and search for appropriate resource nodes
- Three composite behaviors, i.e., *least-loaded move*, *less-loaded move*, and *random move* for agents, which are combinations of the primitive behaviors (i.e., *remain*, *wander*, *join*, and *leave*)
- Each of the above composite behavior is assigned a probability. At a time, an agent probabilistically chooses a behavior to perform
- The service time of a service request is determined by its size as well as the processing speed, corresponding to the service required, of the resource node
- An agent has only local information

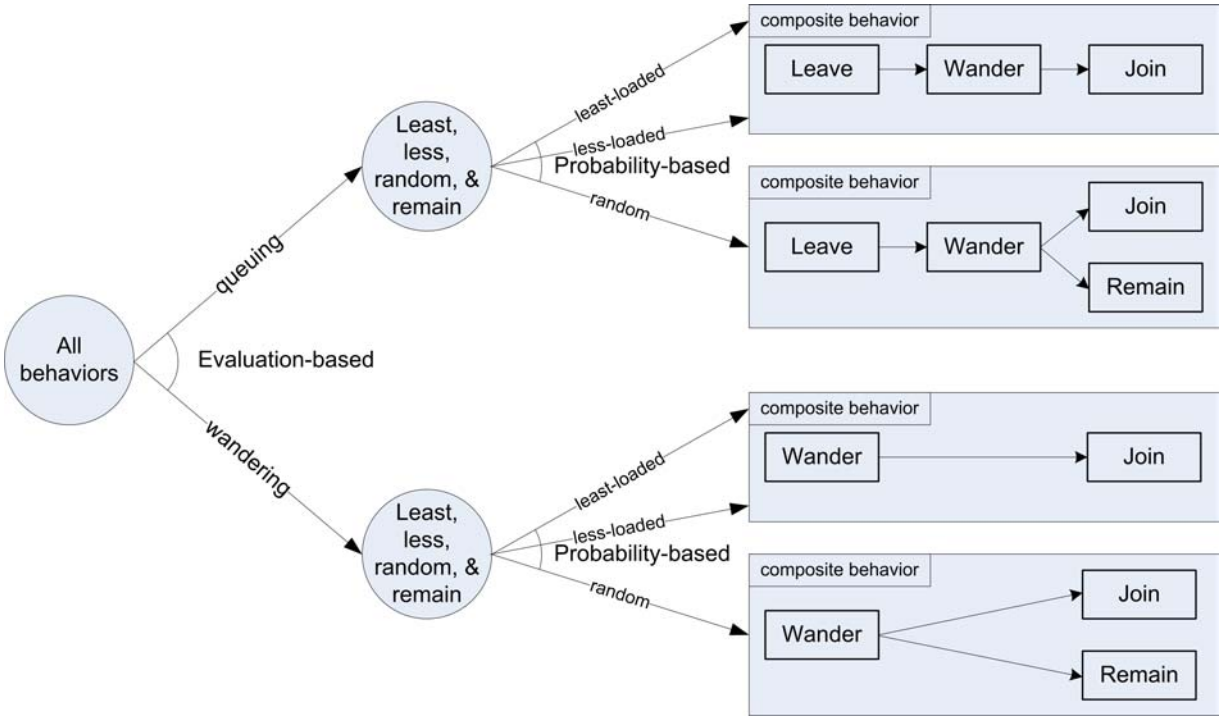
Refined AOC Formulation

- Composite behaviors
  - *Least-loaded move*: moving to a resource node, providing the service required by the agent and with the *least* service request load, in its neighboring region
  - *Less-loaded move*: moving to a resource node, providing the service required by the agent and with the *less* service request load, in its neighboring region
  - *Random move*: randomly moving to a new resource node in its neighboring region

- State description
  - Each agent a has an additional vector: $pcb = \langle p_{least}, p_{less}, p_{random} \rangle$, where p_{least}, p_{less}, and p_{random} denote the probabilities for performing a least-loaded, less-loaded, or random move, respectively
  - p_{least}, p_{less}, and p_{random} are fixed, not updated over time
  - p_{least} and p_{less} should be set relatively large values, while p_{random} should be set a relatively small value

Refined Behavioral Rules

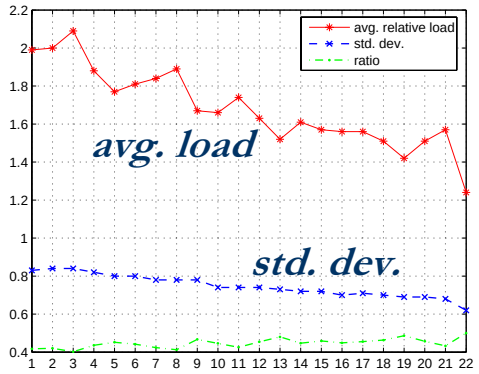
- Behavioral rules

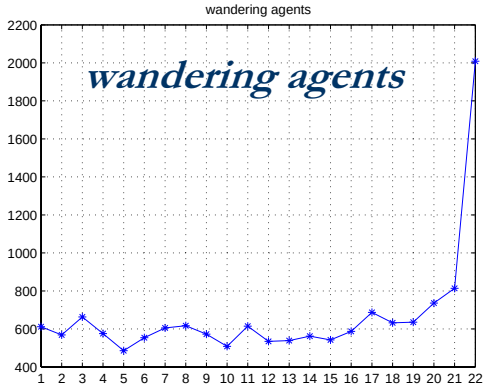


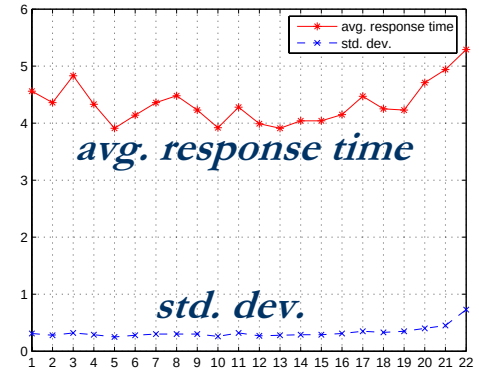
Performance Studies

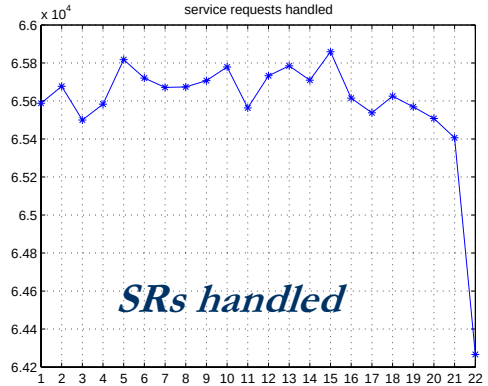
- **II-1.** How does the probability combination affect the performance of the proposed mechanism? Are all composite behaviors necessary?
  - **II-1-A.** Unsaturated situations
  - **II-1-B.** (Approximately) saturated situations
- **II-2.** Whether the proposed mechanism is robust to endure the failures and recovery of resource nodes, and adapt the outcome?
- **II-3.** How does the vision range of agents affect the performance of the proposed mechanism?
- **II-4.** Given two metrics, i.e., absolution and relative service request load, which one is better to load balancing?

II-1-A. Study on: the probability combination in an unsaturated situation







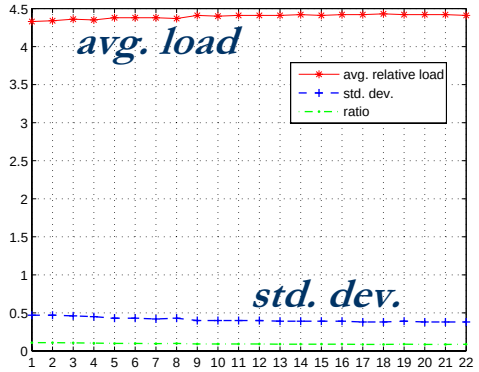


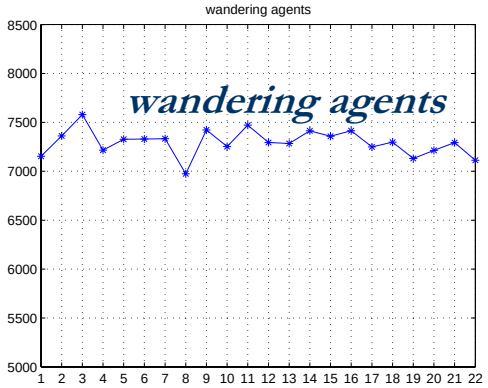
p_{less} ← → p_{least}

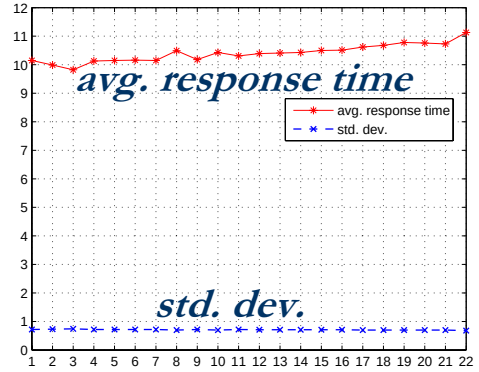
- **Setting:**
  - λ_{iat}=0.75, λ_{ts}=100
  - Processing speeds of services 1 & 2: *200* & *100*
- **Observations**
  - An experiment p_{random} =1.0 :
    - Avg load : 2.5 + Std. Dev. : 3.1
    - The mechanism: effective
  - Large p_{least}, → small std_{rl} → more optimized utilization
    - a large number of wandering agents
    - a low service request load
  - Large p_{less} → large std_{rl} → low degree of resource optimization
    Large p_{less} → relatively short response time → the less-loaded move is necessary

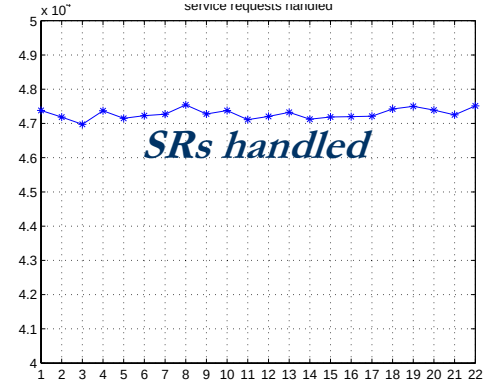
- Random move is also necessary
  - p_{random}= 0 → wandering agents are relatively hard to find suitable resource nodes → a lot of wandering agents
  - Random move helps agents move to new areas such that they can possibly find suitable resource nodes
- In general, in a relatively optimal combination
  - p_{least} and p_{less} : close *0.5*
  - p_{random} : a relatively small, nonzero value, say, *0.01∼0.1*

II-1-B. **Study on:** the probability combination in a saturated situation





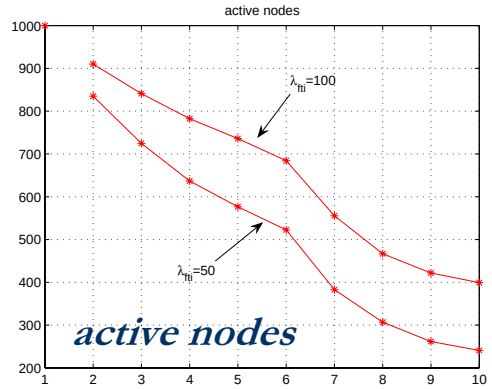


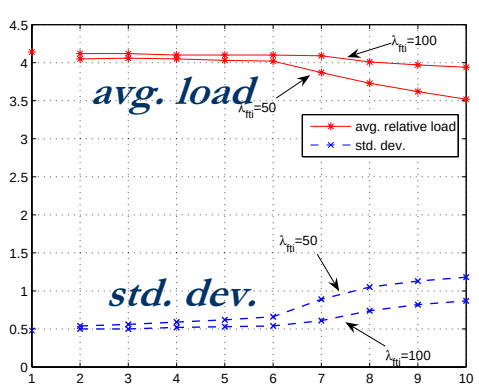


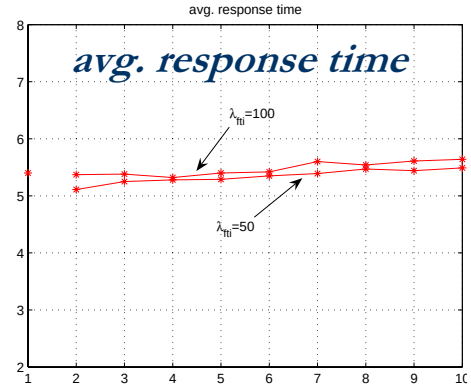
- **Setting:**
  - λ_{iat}=0.45, λ_{ts}=100
  - Processing speeds of services 1 & 2: *200* & *100*
- **Observations**
  - An experiment p_{random} =1.0 :
    - Avg load : 4.5 + Std. Dev. : 2.8
    - The mechanism: effective
  - Random move, not necessary
    - All regions may have the same load situation
    - Least-loaded move or less-loaded move is enough for agents

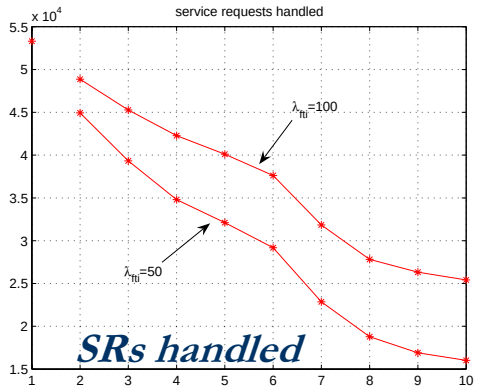
- Different p_{least} and p_{less} ↛ different performance
  - Since a least-loaded move is computationally harder than a less-loaded move, only performing less-loaded move is more reasonable for agents
- In general,
  - If saturated: less-loaded move only
  - If unsaturated: relatively large p_{least} and p_{less}, and relatively small p_{random}

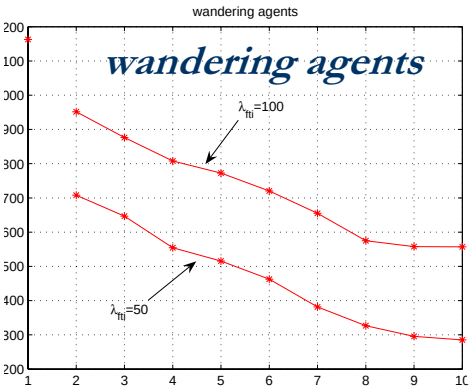
II-2. Study on: Robustness and adaptation







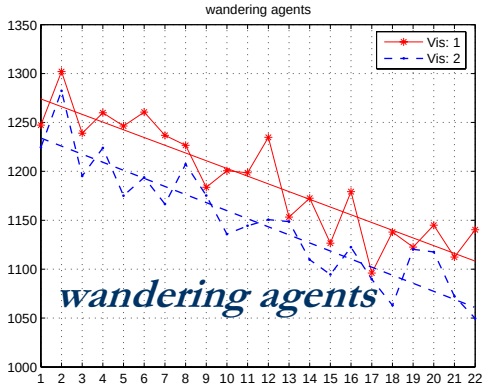


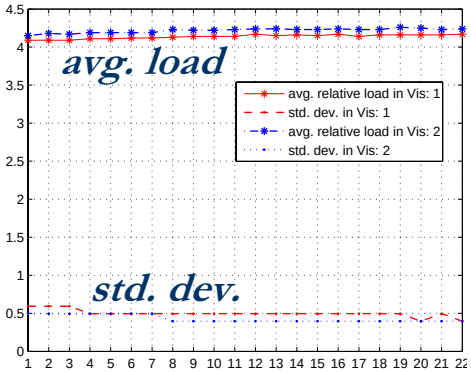


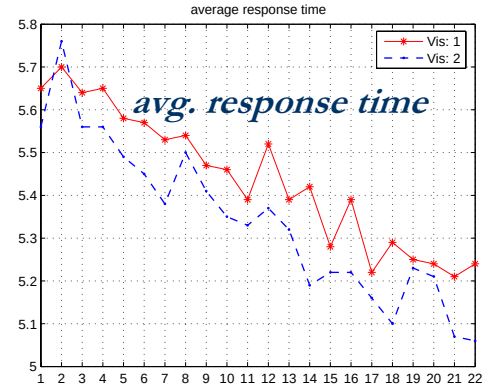
- **Setting:**
  - λ_{iat}=0.45, λ_{ts}=100
#	1	2	3	4	5	6	7	8	9	10
λ_{fti}	–	100	100	100	100	100	100	100	100	100
λ_{rti}	–	10	20	30	40	50	100	150	200	250
λ_{fti}	–	50	50	50	50	50	50	50	50	50
λ_{rti}	–	10	20	30	40	50	100	150	200	250

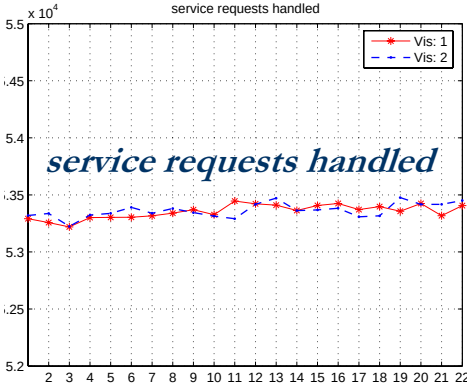
    Note: '–' denotes that in this case, no resource nodes fail and recover.
- **Observations**
  - The proposed mechanism is robust
    - it can endure failures and recovery of resource nodes
  - The effects are mainly determined by the distributions of the f&R time intervals, i.e., λ_{fti} and λ_{rti}
    - $\lambda_{fti}>\lambda_{rti}$: no much effect on the average load
    - $\lambda_{fti}<\lambda_{rt:}$ the effects becomes remarkable: low average load + high standard deviation
    - The smaller the value of λ_{fti} (or, the larger the value of λ_{rti}), the greater the effects
  - The effects on the average response time are not obvious

II-3. Study on: Agents' version range



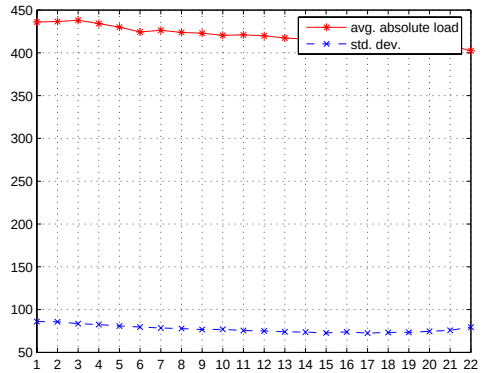


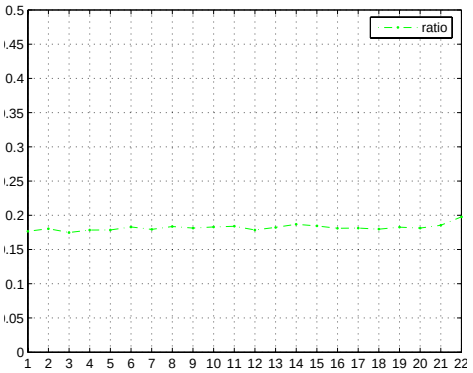


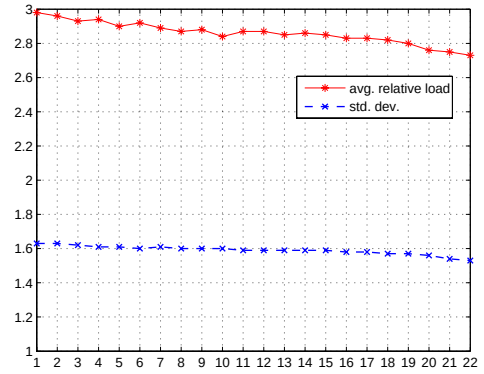


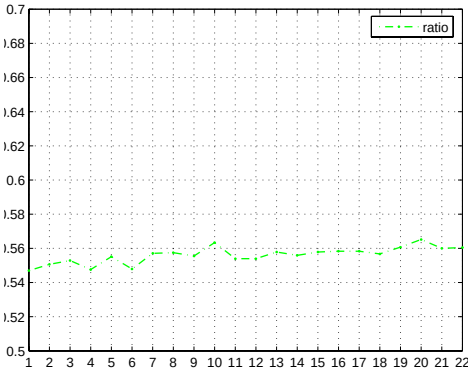
- **Setting:**
  - λ_{iat}=0.45, λ_{ts}=100
  - Vision range: *1* or *2*
- **Observations**
  - The larger the vision range of agents →
    - the more balanced the service request load
    - the higher the average relative service request load
    - the lower its standard deviation
  - From the resource optimization point of view, the larger the vision range of agents, the more optimized the utilization of resources. However, it seems that the effects are not as much as what we expected time of service requests

II-4. Study on: Metrics of agents for selecting resource nodes to move









- **Setting:**
  - λ_{iat}=0.45, λ_{ts}=100
  - Processing speeds of services 1 & 2: *200* & *100*
  - *rn.acp=500* or *rn.rcp=5*
- **Observations**
  - Different metrics (i.e., absolute and relative service request loads) → different performance of the proposed DRO mechanism
  - Given concerning how service requests are distributed to resource nodes such that they have relatively balanced service request load, we should consider:
    - the absolute service request load
    - the processing speeds of resource nodes
  - The relative service request load is a better metrics

Summary

- In an unsaturated resource environment, less-loaded move and random move definitely are necessary. And, the probability combination of agents' behaviors definitely affects the performance of the proposed mechanism. In a good probability combination, p_{least} should be close to p_{less} and p_{random} should be small, but nonzero
- In a saturated resource environment, the probability combination of agents' behaviors has no great effect on the performance of the proposed mechanism. Particularly, least-loaded move and random move are not necessary, and less-loaded move is enough for agents
- Regarding to failures and recovery of resource nodes, if resource nodes can recover quickly from failure (i.e., as compared to the active duration, the failure duration of resource nodes is shorter), failures and recovery of resource nodes will not greatly affect the performance of the proposed mechanism
- As for vision range of agents, the larger the vision range of agents, the better the performance of the proposed mechanism: the higher the average relative service request load, and the lower its standard deviation
- Different metrics of agents for selecting resource nodes have obviously effects on the performance of the proposed mechanism. In the contexts of this thesis, choosing the relative service request load as agents' metric is a better choice, since it considers not only the absolute service request load of resource nodes, but also their processing speeds

Significance, Conclusions and Future work

Significance and Conclusions

- Significance
  - Introduce AOC as a general methodology to WI for addressing the DRO issue
  - Through DRO, re-validate the effectiveness of AOC in solving real-world, large-scale, highly distributed problems
- Conclusions & contribution
  - Surveyed related work on Web Intelligence (WI) and Autonomy Oriented Computing (AOC) (Chapter 2)
  - Presented a brief DRO perspective on WI. Specifically, gave a generalized view of distributed resources on the Web, and described a generalized and abstracted scenario for DRO (Chapter 4)
  - Provided an AOC-based DRO mechanism and the corresponding AOC formulation (Chapter 5)

Significance and Conclusions (Cont.)

- Presented an AOC-based DRO mechanism for homogeneous resource environments and validated it through macroscopical characterization, numerical simulation, and experimentation (Chapter 6)
- Presented an AOC-based DRO mechanism for heterogeneous resource environments and validated it through experimentation (Chapter 7)
- Validated AOC as an effective methodology for distributed resource optimization on the Web in that it satisfies the WI requirements, e.g., adaptive, robust, optimized, etc.. (Chapters 6 & 7)

Future Work

- Service request interdependency
- Agent behavioral variation
- Implementation in a realistic Web environment

Publications

- **Books (3)**
  - Jiming Liu, **Xiaolong Jin**, and Kwok Ching Tsui, *Autonomy Oriented Computing: From Solving Computational Problems to Characterizing Complex Behavior*, Springer, December, 2004, http://www.springeronline.com/sgw/cda/frontpage/0,11855,5-147-72-36093939-0,00.html;
  - Jiming Liu, **Xiaolong Jin**, Shiwu Zhang, and Jianbing Wu, *Multi-Agent Systems: Models and Experimentation* (in Chinese), Tsinghua University Press, November, 2003;
  - Jiming Liu (Author), **Xiaolong Jin** and Shiwu Zhang (Translators), *Multi-Agent Systems: Principles and Techniques* (in Chinese), Tsinghua University Press, November, 2003.
- **Thesis (1)**
  - **Xiaolong Jin**, *Autonomy Oriented Computing (AOC) for Web Intelligence (WI): A Distributed Resource Optimization Perspective*, PhD thesis, Department of Computer Science, Hong Kong Baptist University, March, 2005.
- **Edited Proceedings (1)**
  - **Xiaolong Jin** and Jianliang Xu, eds., *Proceedings of the First HKBU-CSD Postgraduate Research Symposium (PG Day)*, Department of Computer Science, Hong Kong Baptist University, Technical Report COMP-05-002, January, 2005.
- **Invited Book Chapters (5)**
  - **Xiaolong Jin** and Jiming Liu, *Autonomy Oriented Computing (AOC) for Web Intelligence (WI): A Distributed Resource Optimization Perspective*, N. Zhong, J. Liu, eds., Annual Review of Intelligent Informatics, 2005;
  - Jiming Liu, **Xiaolong Jin**, and Kwok Ching Tsui, *Autonomy Oriented Computing (AOC)*, Submitted to the Encyclopedia of Computer Science and Computer Engineering, John Wiley & Sons, 2005;
  - **Xiaolong Jin** and Jiming Liu, *From Individual Based Modeling to Autonomy Oriented Computation*, Matthias Nickles, Michael Rovatsos, and Gerhard Weiss, eds., Agents and Computational Autonomy, LNAI 2969, pp. 151-169, Springer, 2004;

Publications (Cont.)

- **Xiaolong Jin**, Jiming Liu, and Yuanshi Wang, *Agent-Supported WI Infrastructure: Case Studies in Peer-to-Peer Networks*, Y.-Q. Zhang, A. Kandel, T. Y. Lin, and Y. Y. Yao, eds., Computational Web Intelligence: Intelligent Technology for Web Applications, Chapter 24, pp. 515-538, World Scientific Publishing, 2004;
- **Xiaolong Jin** and Jiming Liu, *Agent Networks: Topological and Clustering Characterization*, N. Zhong, J. Liu, eds., Intelligent Technologies for Information Analysis, Chapter 13, pp. 291-310, Springer, 2004.

Journal Papers (6)

- Bingcheng Hu, Jiming Liu, and **Xiaolong Jin**, *Multi-Agent RoboNBA Simulation: From Local Behaviors to Global Characteristics*, Accepted by the Special Issue on Agent-Directed Simulation at Simulation;
- Jiming Liu, **Xiaolong Jin**, and Yuanshi Wang, *Agent-Based Load Balancing on Homogeneous Minigrids: Macroscopic Modeling and Characterization*, IEEE Transactions on Parallel and Distributed Systems, vol. 16, no. 7, pp. 586-598, 2005;
- **Xiaolong Jin** and Jiming Liu, *Characterizing Autonomic Task Distribution and Handling in Grids*, Engineering Applications of Artificial Intelligence, vol. 17, no. 7, pp. 809-823, 2004;
- Jiming Liu, **Xiaolong Jin**, and Kwok Ching Tsui, *Autonomy Oriented Computing (AOC): Formulating Computational Systems with Autonomous Components*, IEEE Transaction On Systems, Man, and Cybernetics – Part A: Systems and Humans (in press);
- Jiming Liu, **Xiaolong Jin**, and Yi Tang, *Multi-Agent Collaborative Service and Distributed Problem Solving*, Cognitive Systems Research, vol. 5, no. 3, pp. 191-206, 2004;
- Jiming Liu, **Xiaolong Jin**, and Jing Han, *Distributed Problem Solving without Communication – An Examination of Computationally Hard Satisfiability Problems*, International Journal of Pattern Recognition and Artificial Intelligence, vol. 16, no. 8, pp. 1041-1064, 2002.

Conference Papers (17)

- **Xiaolong Jin** and Jiming Liu, *Resource Optimization in Heterogeneous Grid Environments*, Submitted to the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), in Compiegne University of Technology, France, September 2005;

Publications (Cont.)

- **Xiaolong Jin** and Jiming Liu, *AOC-Based Load Balancing on Homogeneous Minigrids*, Submitted to the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05), in Compiegne University of Technology, France, September 2005;
- Tingting Wang, Jiming Liu, and **Xiaolong Jin**, *Minority Game Strategies for Dynamic Multi-Agent Role Assignment*, in Proceedings of the 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04), pp. 316-322, Beijing, China, September 2004;
- Bingcheng Hu, Jiming Liu, and **Xiaolong Jin**, *From Local Behaviors to Global Performance in a Multi-Agent System*, in Proceedings of the 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04), pp. 309-315, Beijing, China, September 2004;
- **Xiaolong Jin**, Jiming Liu, and Yuanshi Wang, *Modeling Agent-Based Task Handling in a Peer-to-Peer Grid*, in Proceedings of the 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04), pp. 288-294, Beijing, China, September 2004;
- Yi Tang, Jiming Liu, and **Xiaolong Jin**, *Aggregating Local Behaviors Based upon Lagrange Method*, in Proceedings of the 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04), pp. 413-416, Beijing, China, September 2004;
- **Xiaolong Jin** and Jiming Liu, *The Dynamics of Peer-to-Peer Tasks: An Agent-Based Perspective*, in Proceedings of the Third International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2004), New York, USA, July 2004;
- Long Gan, Jiming Liu, and **Xiaolong Jin**, *Agent-Based, Energy Efficient Routing in Sensor Networks*, in Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS'04), New York, USA, July 2004;
- **Xiaolong Jin** and Jiming Liu, *Properties of Clustering Coefficient in Random Agent Networks* (Excellent Paper Award), in Proceedings of the Second International Conference on Active Media Technology (ICAMT'03), pp. 73-82, Chongqing, China, May 2003;
- Bingcheng Hu, Jiming Liu, and **Xiaolong Jin**, *Phase Transitions in RoboNBA*, in Proceedings of the 5th ACM Postgraduate Research Day (Hong Kong), pp. 73-79, Hong Kong, January 2004;
- **Xiaolong Jin**, Jiming Liu, and Yuanshi Wang, *Modeling Agent-Based Task Handling in a Peer-to-Peer Grid*, in Proceedings of the 5th ACM Postgraduate Research Day (Hong Kong), pp. 87-96, Hong Kong, January 2004;
- Yuanshi Wang, Jiming Liu, and **Xiaolong Jin**, *Modeling Agent-Based Load Balancing with Time Delays*, in Proceedings of the 2003 IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03), pp. 189-195, Halifax, Canada, October 2003;

Publications (Cont.)

- **Xiaolong Jin** and Jiming Liu, *Efficiency of Emergent Constraint Satisfaction in Small-World and Random Networks*, in Proceedings of the 2003 IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03), pp. 304-310, Halifax, Canada, October 2003;
- **Xiaolong Jin** and Hongge Liu, *Research on the Completeness of Pangu Knowledge Base*, in Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (ICMLC'03), Xi'an, China, August 2003;
- **Xiaolong Jin** and Jiming Liu, *Agent Network Topology and Complexity*, in Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS'03), pp. 1020-1021, Melbourne, Australia, July 2003;
- Yi Tang, Jiming Liu and **Xiaolong Jin**, *Adaptive Compromises in Distributed Problem Solving*, in Proceedings of the Fourth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'03), LNCS 2690, pp. 31-40, Springer, Hong Kong, China, March 2003;
- **Xiaolong Jin** and Jiming Liu, *Multiagent SAT (MASSAT): Autonomous Pattern Search in Constrained Domains*, in Proceedings of the Third International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'02), LNCS 2462, Hujun Yin et. al. Eds., pp. 318-328, Springer, Manchester, UK, August 2002;

Technical Reports (4)

- Jiming Liu, **Xiaolong Jin**, and Yuanshi Wang, *Agent-Based Load Balancing on Homogeneous Minigrids: Macroscopic Modeling and Characterization*, Department of Computer Science, Hong Kong Baptist University, Technical Report Comp-04-005, September 2004;
- **Xiaolong Jin**, Jiming Liu, and Yuanshi Wang, *Characterizing the Dynamics of Agent-Based Peer-to-Peer Computing*, Department of Computer Science, Hong Kong Baptist University, Technical Report Comp-04-004, May 2004;
- **Xiaolong Jin** and Jiming Liu, *An Autonomy-Oriented, Distributed Approach to Satisfiability Problems*, Department of Computer Science, Hong Kong Baptist University, Technical Report Comp-04-003, May 2004;
- Jiming Liu, **Xiaolong Jin**, and Kwok Ching Tsui, *Autonomy Oriented Computing (AOC): Formulating Computational Systems with Autonomous Components*, Department of Computer Science, Hong Kong Baptist University, Technical Report Comp-04-001, March 2004.

Thank you!

Q. & A.