Preface

This volume contains the papers presented at SSMCS2015: The 2nd International Workshop on Smart Simulation and Modelling for Complex Systems held on July 25, 2015 in Buenos Aires.

Computer based modelling and simulation has become useful tools to facilitate humans to understand systems in different domains, such as physics, astrophysics, chemistry, biology, economics, engineering and social science. A complex system is featured with a large number of interacting components (agents, processes, etc.), whose aggregate activities are nonlinear and self-organized. Complex systems are hard to be simulated or modelled by using traditional computational approaches due to complex relationships among system components, distributed features of resources, and dynamics of environments. Meanwhile, smart systems such as multi-agent systems have demonstrated advantages and great potentials in modelling and simulating complex systems. The 2nd International Workshop on Smart Simulation and Modelling for Complex Systems (SSMCS2015) aims to bring together researchers in both artificial intelligence and system modelling/simulation to discuss research challenges and cutting edge techniques in smart simulation and modelling.

The topics of SSMCS include but are not limited to:

- Agent based simulation for complex systems
- Agent based modelling for complex systems
- Large-scale simulations
- Network simulation and modelling
- Environment and ecosystem simulation and modelling
- Smart Grid/Service simulation and modelling
- Simulation of social and economic organizations
- Simulation of social complexity
- Cooperation, coordination, negotiation and self-organisation in complex systems
- Market-based model and simulation
- Transportation model and simulation
- Crowd model and simulation
- Evacuation model and simulation
- Human behaviour modelling, learning and simulations

Special thanks to all PC members and paper reviewers for their valuable contributions to the workshop. We also thank all supports from all authors and workshop participators.

July, 2015

SSMCS2015 Organisation Committee
Dr. Quan Bai
Dr. Fenghui Ren
Prof. Minjie Zhang
Prof. Takayuki Ito
Table of Contents

Adaptive Forwarder Selection for Distributed Wireless Sensor Networks . . 1
Nor Khalid and Quan Bai

Trust Transference on Social Exchanges among Triads of Agents Based
on Dependence Relations and Reputation .............................. 2
Yaneeda Rojas, Diana Francisca Adamatti and Graçaliz Dimuro

A Multiagent-Based Domain Transportation Approach for Optimal
Resource Allocation in Emergency Management ...................... 3
Jihang Zhang, Minjie Zhang, Fenghui Ren and Jiakun Liu

A proto-type of a portable ad hoc simple water gauge and real world
evaluation ............................................................... 4
Takanobu Otsuka, Yoshitaka Torii and Takayuki Ito

Exploiting Vagueness for Multi-Agent Consensus .................... 5
Michael Crosscombe and Jonathan Lawry

Selecting Robust Strategies Based on Abstracted Game Models ....... 6
Oscar Veliz and Christopher Kiekintveld

Simulating and Modeling Dual Market Segmentation Using PSA
Framework ............................................................... 7
Quan Bai, Jiamou Liu and Ziheng Wei

CORPNET: Towards a Decision Support System for Organizational
Network Analysis using Multiplex Interpersonal Relations .......... 8
Jiamou Liu, Anastasia Moskvina and Michael Ourednik
Program Committee

Quan Bai  
Shantanu Chakraborty  
Enrique De La Hoz  
Katsuhide Fujita  
Naoki Fukuta  
Tomomichi Hayakawa  
Takayuki Ito  
Jiamou Liu  
Qing Liu  
Xudong Luo  
Iván Marsa Maestre  
Fenghui Ren  
Xijin Tang  
Guandong Xu  
Dayong Ye  
Chao Yu  
Minjie Zhang  

Auckland University of Technology  
NEC Corporation  
University of Alcalá  
Tokyo University of Agriculture and Technology  
Shizuoka University  
Nagoya Institute of Technology  
Nagoya Institute of Technology  
Auckland University of Technology  
CSIRO  
Sun Yat-sen University  
University of Alcalá  
University of Wollongong  
CAS Academy of Mathematics & Systems Science  
University of Technology Sydney  
Swinburne University of Technology  
Dalian University of Technology  
University of Wollongong
Additional Reviewers

Wang, Xishun
Zhan, Jicyu
Adaptive Forwarder Selection for Distributed Wireless Sensor Networks

Nor Azimah Khalid and Quan Bai

Abstract Wireless Sensor Network has emerged as a promising networking technique for various applications. Due to its specific characteristics, such as non-rechargeable, low-power multi-functional sensor nodes, limited sensing, computation and communication capabilities, it is challenging to build networking protocols for Wireless Sensor Networks. In this paper, the focus is on addressing the routing issue with regards to energy efficiency and network lifetime. An adaptive and self-organized routing protocol for distributed and decentralized network, called Distributed Adaptive Forwarder Selection, is proposed. Multiple factors, involving cross layers were used for selecting the adequate forwarders for packets. Our approach is dynamic as no fix topology or static role assignment is required. In addition, our approach is flexible where Receiver can make decision based on its current capabilities. We have performed simulations of our proposed protocol and compared with two existing routing protocols in terms of node lifetime, average energy consumption and average residual energy. The results show that our proposed protocol performed better than LEACH and MOECS.

1 Introduction

In general, Wireless Sensor Network (WSN) is a wireless network which consists of large numbers (hundreds to thousands) of irreplaceable and low-power multi-functional sensor nodes, operating in an unattended environment with limited sensing, computation and communication capabilities [1] used in a wide range of applications. Physical resources are a critical element for any devices which in communication devices, the resources include the memory, processing power and energy.
These elements have been recognized as constraints on wireless sensor nodes capabilities due to the size of the sensor node itself (Karl and Willig, 2007). A number of previous works have focused on these constraints for designing the communication and information processing elements for wireless sensor networks.

Communication and information processing were identified as crucial processes in wireless sensor networks environment which are related with the constraints mentioned. Research in communication elements have been focusing on managing communication resources for data delivery processes. Communication process has been identified as highly resource consuming especially in energy where the data transmission contributes high energy consumption if the resource is not well managed (Giuseppe et al., 2009). There are two elements of communication for wireless sensor networks application: routing mechanism and media access control (MAC) protocol. In previous studies, several previous works for energy perspective issues have been briefly described on both elements. This involves reducing number of transmissions and distance of transmission via clustering or scheduling mechanisms at specific network layers.

For large scale networks, decentralized architectures are more appropriate as high transmission cost and delay might involve, especially if the central controller is located far away. Furthermore, in a centralized architecture, if the central node fails, then the entire network will collapse. On the other hand, decentralized control architecture are more reliable for large networks and can provide better collection of data and backup, in case of failure of the central node. However, decentralized approach is very challenging in terms of topology establishment and re-establishment especially in unaccessible applications such as battlefield and disaster management. These non replenish nodes need to self-organized themselves to sustain longer. This requires nodes in decentralized scheme to adapt accordingly with dynamic changes of environment (i.e. the network topology) and self-configure themselves without human intervention.

Shortest path algorithms based on either hop count or energy consumption is typically employed in routing protocol of ad hoc networks to achieve high energy efficiency [10]. However, relying on these parameters might cause hot spot scenario, i.e, sensor nodes that are frequently used might get depleted. In such case, a more reliable nodes should be selected. These lead to the need of having a more adaptive parameters to be considered during path selection. Furthermore, node preferences might be different, i.e. one node will choose a path which is nearer to it, which could be far to other nodes.

In this paper, we focus on selection of nodes to relay packets (i.e., Forwarder) and propose an adaptive forwarder selection approach for distributed WSNs. The Distributed Adaptive Forwarder Selection (DAFS), comprises three phases i.e., Eligibility Determination, Forwarder Selection and Receiver Acceptance. In DAFS, multi-criterion parameters which comprises energy, distance and buffer size are considered in reward functions. The proposed approach is adaptive and suitable for dynamic environments, where nodes actions are determined based on their current capabilities and state of environment. In addition, it is suitable for largely distributed networks, where only decentralized approach is feasible.
The paper is organized as follow. In Section 2, we describe our problem description and definitions. Section 3 introduces the protocol applied in the proposed approach. Section 4 explains our propose model. Section 5 presents the simulation results. Finally, conclusion is presented in Section 6.

2 Related Work

In this section, we briefly review related work on energy-efficient routing protocols and learning-based protocols. Most of the previous studies have shown that clustering approach can reduce the energy expenditure when merging all the information from the nodes into one cluster head which is responsible to process it and deliver it to the sink or base station. The limitation of sensor nodes usage for processing information has given a better energy consumption management which results to the sensor nodes lifetime to increase.

One of the well-known cluster-based protocols is Low Energy Adaptive Clustering Hierarchy (LEACH) [6]. In LEACH, data collection is performed periodically, which involves two phases, i.e. Cluster Head (CH) selection and cluster formation. The selection of CH in LEACH is based on closest distance. Each CH creates Time Division Multiple Access (TDMA) schedule for member nodes. CH also select code division multiple access (CDMA) to reduce inter-cluster interference. Members will collect information and use their allocated TDMA slots to transmit their collected data to CH. In [3], an energy efficient cluster formation algorithm (MOECS), was proposed based on a multi-criterion optimization technique. The selection of cluster heads is restricted to certain optimal value (i.e. optimal radius and distance from normal nodes).

Learning-based approach such as Reinforcement Learning (branch of machine learning), is commonly used in distributed system [4]. Distributed independent reinforcement learning (DIRL) for example, is based on independent learning, i.e., each agent can autonomously and dynamically self-configure in order to maximize its own reward [13]. A reward-based dynamic approach based on two tier reinforcement learning scheme (micro learning and macro learning) were proposed in [12]. In their approach, an individual node was able to self-schedule its task using its local information and through learning [11].

Q-learning is a model-free RL technique, based on agents taking actions and receiving rewards from the environment in response to those actions [5]. In [8], Dimarogonas and Johansson proposed a combinatorial reverse auction that operates in two phases using RL and some economic models for energy optimization in sensor networks. The Q value was represented an estimate cost of the route through neighbor comprised of hop count (account for energy efficiency) and minimum battery level among nodes. In [7], the potential of using energy aware metrics in RL based routing algorithms for WSN was studied, combining energy aware metrics with load balancing metrics. In [10], a machine-learning-based routing protocol for energy-efficient and lifetime-extended for UWSN, i.e. QELAR, was proposed. In QELAR,
residual energy of each node and energy distribution among a group of nodes were used in it’s lifetime-aware reward function, for calculating the Q-value (in selecting forwarder for packets). In [9], role-free clustering assignment was combined with learning dynamic network properties such as battery reserves. Less energy was consumed by using machine learning to enable nodes to independently decide whether or not to act as a cluster head on a per-packet basis in comparison to a traditional approach.

2.1 Problem Description and Definitions

In this paper, we propose an adaptive, energy efficient, and lifetime-aware Forwarder selection approach, based on Q-learning technique. Using an action-value function (Q-value), which gives the expected reward of taking an action in a given state, the distributed learning agent is able to make a decision automatically. The proposed approach has the following features:

- Dynamic Network: In largely distributed network, link quality is not guaranteed. Link failure due to node’s energy depletion, causes topology changes. Using Q-learning algorithm, selection of alternative link is possible as node selects next best forwarder based on current situations.
- Adaptive: We define our node role as Forwarder, Receiver and normal nodes. Node will decide on its role based on its present capabilities. It is adaptive to available resources i.e a Receiver can accept packets that they are capable to process as that time accept more when less busy, a node can be Forwarder at a time but not at the other time if there is other more capable node to forward the packet etc.).
- General Framework: Q-learning behaviors is determines by its reward function. We proposed a flexible and dynamic approach for the nodes to react based on its present capabilities.
- Load Balancing: Less energy is consumes when choosing a path based on shortest path. However, this may cause the link failure as choosing the same node to forward packets could drain its energy faster. We consider multiple parameters to allow alternative path selection.

2.1.1 Definitions

The network in our model is consider as a complex system comprises of nodes, called agents. Throughout this paper, the term, node and agent is use interchangeably.

**Definition 1:** A WSN is defines as connected undirected weighted graph $G = (V, E)$, where $V$ is group in the network comprises of agents, i.e. $V = a_0, a_1, a_2, ... a_n$. 
\( E = e_1, e_2, \ldots, e_m \) is a set of edge in group. The edge, \( e_k = (a_i, a_j) \) denotes the communication links between sensor \( a_i \) and sensor \( a_j \) (they are in each other’s radio transmission range).

**Definition 2:** Each agent \( a_{ai} \) is defined as \( a_{ai} \in (Ra_{ai}, Act_{ai}, Ca_{ai}, R_{wd{i}}) \), where \( Ca_{ai} = (E_{init{a_i}}, E_{res{a_i}}, E_{Neigh{a_i}}, Dist, BF_{a_{ai}}, BF_{Neigh{a_i}}) \); \( E_{init{a_i}} \) is \( a_i \) initial energy, \( E_{res{a_i}} \) is \( a_i \) residual energy, \( E_{Neigh{a_i}} \) is \( a_i \) immediate neighbors’ energy, \( Dist \) is distance between agents. \( BF_{a_{ai}} \) is current buffer size of \( a_i \), \( BF_{Neigh{a_i}} \) is current buffer size of \( a_i \) neighbors, \( Ra_{ai} \) is \( a_i \) Role, \( Act_{ai} \) is action that \( a_i \) takes and \( R_{wd{i}} \) is Reward that \( a_i \) gains (which is explain in Definition 3, 4 and 5 respectively).

**Definition 3:** There are three types of Role, i.e., R in the proposed model, which are Forwarder, Receiver and Normal node. In dynamic environment such as WSN, it is not practical to select a node as Forwarder permanently, as it will cause the node to die faster. We propose a more distributed approach which allow flexibility in being Forwarder, based on current capabilities. A Forwarder can be a normal node at other time when its energy has degraded or it is currently processing many tasks.

**Definition 4:** An action set \( Act_{ai} \) is defined as: \( Act_{ai} = (act_1, act_2, \ldots, act_n) \), where \( (act_1, act_2, \ldots, act_n) \) are possible actions that \( a_i \) can perform. As Forwarder, below actions are possible:

- Forwarding packets received from one agent to another agent.
- Discard packets if no Forwarder is identified.

As a Receiver, there are three possible actions:

- Accept packets based on current capabilities.
- Reject packets if buffer is full (currently busy).

**Definition 5:** Reward function \( R_{wd{i}} \), represents expected reward received by agents when transiting from one agent to another agent. The goal of our algorithm is to get the packet delivered from one agent to another agent, with maximum reward, i.e., minimum cost. The reward function is as in Equation 2-4.

### 3 System Framework and Interaction Protocol

There are three modules in the proposed approach, which are Eligibility Determination Module, Dynamic Forwarder Selection Module and Receiver Acceptance Module. Algorithm 1 shows the steps involved. When an agent has packet to transfer, they will compare among eligible neighbors, which one is the most capable (i.e. the one having the highest Q value). If no agent can accept the packet, after timeout, it will drop the packet. Among Eligible agents, once they receive packets, they will decides whether to accept or reject the packets. The amount of packets it will accept is depending on its current capabilities i.e. accept packets that they are able to process.
Algorithm 1: Forwarder Selection.

begin
   for each step of episode do
      Prior to any decision, individual agent will share its information $E_{\text{res}}$, $\text{Dist}$, $BF_{a}$
      and $R$ with its neighbors;
      if current $BF < \text{maximum BF AND } E_{\text{res}} > \text{Minimum Energy}$
      then
         Decrease TIMEOUT ——(as it causes delay)
         Set Status as Eligible Forwarder
         Accept packet according to available BF
      ELSE
         if $BF > \text{maximum BF OR FULL OR } E_{\text{res}} < \text{Minimum Energy}$
         then
            Set Status as Not Eligible
            Choose Forwarder that give $Q_{\text{max}}$

3.1 Eligibility Determination

In Eligibility Determination module, an agent decides whether to be a Forwarder
or not, based on its current capabilities (see Definition 2). The congestion or queue
between receives/transmits will determine agent eligibility at the local level. A fully
occupied buffer indicates agent is not capable to process any information at that
particular time. The residual energy indicates eligibility at higher level, towards
wider context i.e network layer. Agent that decided to be Forwarder will inform it’s
neighbors about it’s decision. In Dynamic Forwarder Selection, agent having more
than one potential Forwarder will select the best Forwarder to forward packets based
on $Q$ value. Forwarder having the highest $Q_{\text{max}}$ will be chosen. $Q$ value is explained
in Section 3.2. When Forwarder receives packets from neighbors, it will process the
received packets, according to its current capabilities.

3.2 Forwarder Selection

To assist agent in selection decision, we use Q-learning approach where using this
approach, agent tends to select Forwarder that gives maximum $Q$-value. The $Q$-
value used not only considers the successful transmission but also the fail transmis-
sion value. Our approach is not only concerns on selecting the best Forwarder but
also allow Forwarder to negotiate as they wish. This scenario is considered as a two
way directional selection. The expected reward that can be received by taking an
action at time $t$ and the state at time $t$ is denotes as:

$$Q(s_t, A_{ct}) = R_{\text{wdtotal}} + \gamma \sum_{s_t, s_{t+1}} P^a \max Q(s_{t+1}, a)$$  

γ in Equation 1 is the discount factor in the range of (0, 1). When γ is set to 0, the system only considers the current reward and it acts similarly to a greedy algorithm. When γ is set to 1, the system will strive for a long-term high reward. As explained in Section 2, agent’s capabilities are evaluated when determining Eligibility as Forwarder. It is also used as input in reward functions, which is then applied in Q value calculation. We defined two reward functions, comprises \( R_{wd \text{ success}} \) as in Equation 2 and \( R_{wd \text{ fail}} \) as in Equation 3. If the packet forwarding attempt from \( a_i \) to \( a_j \) is successful, the reward function is defined as:

\[
R_{wd \text{ success}} = -g - a_1[c(a_i) + c(a_j)] + \text{distance}
\]

(2)

where \( c(a_i) = 1 - \frac{\text{Energy}_{\text{res}}(a_i)}{\text{Energy}_{\text{init}}(a_i)} \); \( a_i \) is node i, \( a_j \) is node j, distance is between node i and node j and \( a_1 \) is less than 1. On the other hand, if the packet forwarding attempt from \( a_i \) to \( a_j \) fails, the reward function is defined as:

\[
R_{wd \text{ fail}} = -g - b_1c(s_n) + \text{distance}
\]

(3)

where \( b_1 \) weight that can be tuned 0.5; \( g \) is constant punishment or cost when node try to forward packet. The total reward, \( R_{wd \text{ total}} \) total gain by agent is:

\[
R_{wd \text{ total}} = R_{wd \text{ success}} + R_{wd \text{ fail}}
\]

(4)

\( R_{wd \text{ total}} \) is use in Q value calculation (Equation 1) above. The far an agent from other agent is, the more energy is consumes for transmission. Thus, it will choose Forwarder that is nearer to it.

### 3.3 Receiver Acceptance

In the Receiver Acceptance module, upon receiving a packet, agent will check its current processing task. It will accept packet according to its current capabilities, i.e. if it is currently processing certain task but still have available buffer, it will accept an amount of packets based on its remaining buffer.

### 4 The Distributed Adaptive Forwarder Selection (DAFS)

Many energy efficient and lifetime-aware approaches proposed solutions either at Physical layer, MAC layer, Network layer, Transport or Application layer. Even though such solutions can improve network performances in terms of network lifetime, energy efficiency, power consumptions etc., both analytical studies and experimental works in WSN highlight the important interactions between different layers of the network stack [2]. In this research, we consider multi-variables parameters involving Network layer, MAC layer and as well distance between nodes. In this
section, we will elaborate on those parameters, which are used in our reward functions.

4.1 Multi-variables Parameters

We consider agent capabilities as Energy, Buffer Size and Distance. For most applications, a wireless sensor node is not replenish. Therefore, there is strong dependence on battery lifetime. Similar to traditional Network layer, data transmission is linked to data communication area, which relates to certain layer; the link layer or MAC layer, Network layer (routing protocols) and Transport layer (transport protocol). Below are multiple factors considered in our approach:

4.1.1 Communication Energy

The main task of sensor node is to detect events, perform local processing and transmit the data. Power consumption can be divided into sensing, communication and data processing. In decentralized network, nodes may need to know its neighbors’ latest state. However, in such network, continuous updates will require a lot of energy. We minimize such energy consumption by allowing only effected nodes to update and updates will only be sent if there is changes (i.e. if its energy is depleted an reaching a threshold value or if there is topology change, such as a new node joining the network). Hence, our concern is on communication energy as sensor node expends the maximum during this phase (transmitting and receiving data). The energy model in [3] is adopted here.

\[ E_{TX} = \ell \times E_{elect} + \ell \times \varepsilon_{fs} \times d^2 \]  

(5)

The energy expended in receiving an \( \ell \)-bit message is given by,

\[ E_{RX} = \ell \times E_{elect} \]

(6)

where \( E_{elect} \) is the base energy required to run the transmitter or receiver circuitry (50nJ/bits), \( \varepsilon_{fs} \) is the energy consumed in an amplifier (10pJ/bits/m²) and \( \ell \) is the length of message (4000bits).

4.1.2 Local Congestion Control - MAC Layer Solutions

The second issue considered is concerning local congestion, by limiting the traffic that an agent can relay. An agent may participate in the communication if it can relay
the packet which is based on its communication activity. For this reason, buffer size is considered as another important factor in the proposed model, i.e., when packets arrive, they have to be processed and transmitted. If packets arrive faster than the agent can process them, the agent puts them into the buffer until it can get around to transmit them. The maximum queuing delay is proportional to buffer size. The longer the line of packets waiting to be transmitted, the longer the average waiting time is. The queue of packets waiting to be sent also introduces a potential cause of packet loss. Since the agent has a finite amount of buffer memory to hold the queue, an agent which receives packets at too high rate may experience a full queue where the agent has to simply discard excess packets.

4.1.3 Distance

In some cases, agents may be located far away from each other or from the Sink. Direct communication or peer-to-peer communication between nodes, especially in large distributed area is impossible, as it causes higher transmission cost and deplete faster. Thus, we consider distance as another important parameter. For example, if there are two Forwarders that is within agent’s proximity, where Forwarder A having more energy and less buffer, the agent might choose Forwarder B, which has less energy and buffer compared to Forwarder A but is nearer to it, taking into account, the significant energy consumption for longer distance communication.

5 Simulation Results

In this section, we evaluate DAFS by comparing it with two cluster-based approaches, i.e., LEACH and MOECS. Two metrics are used to measure the performance of different protocols: first node death time and average residual energy. The first metric needs to be maximized, while second metric needs to be minimized. First node death time is the time when the battery of the first sensor node is depleted. Each sensor node has the goal of maximizing its own packet delivery to destination (that is to avoid packet loss by sending only to forwarder that is the most capable). Table 1 provides the common simulation parameters, which is also used in our experiments. Network lifetime is the most important performance metric for WSNs. Using this metric, DAFS, LEACH and MOECS protocols are evaluated. The nodes in each simulation are distributed in a 100 $\times$ 100 $m^2$ region, where the location of nodes are selected randomly and that no two points have the same location. The Sink is given a fixed location. All the nodes are homogeneous and have the same capability.

Figure 1a shows the results of the first node death (round number) for two different network sizes. The first node death for network size 200 nodes, occurs at 710 rounds in LEACH, at 920 rounds in MOECS and at 3940 rounds in DAFS. While for network size 500 nodes, the first death round occurs at 730 rounds in LEACH, at
980 rounds in MOECS and at 3472 rounds in DAFS. This might be due to communication involves during clustering phase in LEACH and MOECS. In addition, as more criterion is considered in DAFS, i.e., including nodes buffer size allows nodes to choose other alternative Forwarder.

In DAFS, multiple parameters that influence energy consumption were included.

These parameters include communication cost from sensor node to the Forwarder, communication cost from Forwarder to the Sink, and the Forwarder’s residual energy, which help sensor nodes achieve balanced energy dissipation in the system.

Figure 1b depicts the results for average energy consumed per round for two different network sizes using random topology which shows that our DAFS approach performs better than the other two. In addition to the balanced energy dissipation behaviors, such as distance, helps DAFS achieves minimum energy consumption compared to LEACH due to MOECS.

Figure 2a shows number of alive nodes in the network after 5000 rounds where nodes in DAFS survives much longer compared to the other two. Figure 2b illus-

---

**Table 1** Simulation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>100-500</td>
</tr>
<tr>
<td>Deployment Area</td>
<td>100 x 100 m</td>
</tr>
<tr>
<td>Data Packet Size</td>
<td>500 bytes</td>
</tr>
<tr>
<td>Control Packet Size</td>
<td>25 bytes</td>
</tr>
<tr>
<td>$E_{Elect}$</td>
<td>50nJ/bit</td>
</tr>
<tr>
<td>$\varepsilon_{fs}$</td>
<td>10pJ/bit/m^2</td>
</tr>
<tr>
<td>$\varepsilon_{mp}$</td>
<td>0.0013J/bit/m^4</td>
</tr>
<tr>
<td>Initial Energy for sensor node</td>
<td>0.5J</td>
</tr>
<tr>
<td>Network Topology</td>
<td>Random</td>
</tr>
</tbody>
</table>

---
Fig. 2 (a) Number of alive nodes for 5000 rounds. (b) Average Energy consumed per round in DAFS, LEACH and MOECS.

trates results for the random topology where y-axis indicates the average residual energy and x-axis denotes the number of rounds. The residual energy of the system can also provides estimation of the network life. It can be observed that the mean residual energy of the system in the case of DAFS is higher than that of the other protocols. Hence, the network life under DAFS is enhanced compared to LEACH and MOECS. Unlike these cluster-based approaches (LEACH and MOECS), our approach did not involve cluster formation phases and is a distributed approach, as selection of Forwarder is based on learning i.e., the $Q_{\text{max}}$ value.

6 Conclusion and Future Work

As a resource constraint node, the use of sensor nodes in large scale network has posed challenges in terms of energy efficiency and decentralized approach. This challenges can be overcome by ensuring energy is not use unnecessarily in transmission (multiple redundant packet, frequent use of same nodes etc.) Thus, the selection of relay node, i.e., Forwarder, is crucial.

Decentralized architectures are more appropriate in many WSN applications. However, without the present of central controller, node needs to make its own decision based on limited information. In this paper, we consider multi-criterion parameters in Forwarder selection and assists nodes decision by using a distributed learning-based approach. Our solution is adaptive as it is based on agent’s current capabilities, that are changing dynamically when it gets depleted etc. With this technique, it is possible to consider multiple individual metrics for forwarder selection which is critical for well balanced energy dissipation of the system.

Simulation results demonstrate that DAFS achieves significant energy savings
and enhances network lifetime when compared to LEACH and MOECS protocols. Multiple parameters involved in forwarder selection process for DAFS help to dissipate energy at a much more balanced rate as compared to other protocols and also it shows that the ability of DAFS to scale both from the network deployment area and node density makes it viable energy efficient schemes for WSNs.

References

Trust Transference on Social Exchanges among Triads of Agents Based on Dependence Relations and Reputation

Yunevda E. L. Rojas, Diana F. Adamatti, and Gracaliz P. Dimuro
Universidade Federal do Rio Grande - FURG
Centro de Ciências Computacionais
Rio Grande / RS, Brasil
{yuniekita,dianaada,gracaliz}@gmail.com

Abstract. This paper proposes the addition of a third agent to the model proposed by Piaget to Social Exchanges. This agent, called Intermediary Agent, may outsource a service in favor of another agent. This situation can be used to develop a social exchange model for triads of agents in order to evaluate the process of trust transfer between agents who are not familiar with each other, i.e., they have never interacted directly, just by the intermediary agent. The trust transfer analysis is based on exchange value, reputation and dependence relationship concepts, all of which are analyzed interdependently. This way, it complements the Piaget model, generating a contribution to the study of non-economic aspects of the exchange process. The generated information can be used to guide agents’ decisions about partner choice in future exchanges. Additionally, it provides a starting point to understanding the agents’ behavior in interactions among more than three agents.

1 Introduction

On the field of the multi-agent systems, the analysis of human society interactions models and theories shows that there are many ways of interacting, that can happen in different levels. For example, it can be an exchange of information, a negotiation, a discussion, the development of shared visions about an environment, or even the formation or dissolution of organizational structures.

The theory developed by [9] is widely used in this field of studies. It models a social exchange as a service exchange interaction between a pair of agents. After each exchange, an evaluation is made by the involved agents, generating the exchange values concept, divided in investment, satisfaction, credit and debit, which are qualitative values. These values help the agents to choose partners for future exchanges. Many studies refer to these interactions, focusing in different points, like coalitions, task delegation, cooperation and coordination between agents [8] [4]. This collective behavior of the agents is used to solve problems in multi-agent systems [13].

Thus, the analysis of the exchange values, together with the relations of dependency, reputation and trust in the approach to social exchanges, begins,
the possibility to of understanding the subjective values, that are part of the exchange process, of qualitative nature and non-economic (good, bad, trust, etc.), with which everyone judges the daily exchanges he which usually cannot be faithfully presented quantitatively, due to the lack of neat objective conditions for their measurement. This, in comparison to the classical exchange, based on economic values, quantitative utility functions based approach. For instance, the amount of utility to be obtained from an exchange.

In this paper, the concept defined by Piaget is expanded for social relations composed by exchanges among three agents, by adding an Intermediary Agent, that acts as both requester and supplier of a service. The intermediary agent is characterized by its impossibility of performing a service, requested as payment for a debit, but compromised to fulfill its duty with the requester, searches for a third agent that is able to perform the requested service, outsourcing the request. To keep the continuity of the social exchange process, it is needed to guarantee the existence of future exchanges and the correct choice of exchange partners. To achieve that goal, besides the exchange values, other concepts are needed, that are the dependence relation between agents and the reputation of each involved agent. Considering that a dependence relation, as stated by [11], happens when an agent wants to achieve certain state, which is its objective, but it does not have the possibilities to achieve. So, it searches for a second agent who has every condition needed to help it achieve its goal. The actions of the second agent will become a resource to make it possible for the first agent to achieve its objective. [10] study the dependence relations in more details. Reputation is seen as a social tool that has the objective to reduce the uncertainty about unknown individuals before interacting with them, as stated by [3] [2] [1] [7] [6] have done studies in more details. Finally, this work integrates within the multi-agent systems, the basis of social exchange, dependence, reputation and trust, by treating them interdependently, never alone. Thus, it allows the development of a social exchange model for triads of agents to evaluate the process of trust transference.

2 Proposed Models

Four models will be presented: social exchange with intermediary agent; dependence relationship: reputation; and trust transfer. They are described by equations which integrate sequentially.

2.1 Social Exchange with Intermediary Agent (SEIA)

The proposed model, entitled social exchange with intermediary agent (SEIA), consider three agents, $X$, $I$ and $Y$, with independent characters. Both agent $X$ and agent $Y$ know and they are related with agent $I$, but they do not know each other. These agents interact in two exchange stages.

Stage I is composed by two independent process of exchange. In Exchange 1, agent $X$ performs a service for agent $I$. In Exchange 2, agent $I$ performs a service for agent $Y$. Each interaction generates exchange values, as shown in Figure 1a.
- **Stage I - Exchange 1:**
  1. $R_{xi}$: Investment value of agent $X$ for accomplishing a service for agent $I$.
  2. $S_{ix}$: Satisfaction value of agent $I$ for the service received from agent $X$.
  3. $T_{ix}$: Debt value of agent $I$ from the satisfaction about the service received from $X$.
  4. $V_{xi}$: Credit value that agent $X$ acquired with agent $I$ for the accomplishment of a service.

- **Stage I - Exchange 2:**
  1. $R_{iy}$: Investment value of agent $I$ for accomplishing a service for agent $Y$.
  2. $S_{yi}$: Satisfaction value of agent $Y$ for the service received from agent $I$.
  3. $T_{yi}$: Debt value of agent $Y$ from the satisfaction about the service received from $I$.
  4. $V_{iy}$: Credit value that agent $I$ acquired with agent $Y$ for the accomplishment of a service.

In Stage II, agent $X$ charges a debt from agent $I$. This charge involves a service concerning its virtual value of credit ($V_{xi}$), obtained in Stage I. Agent $I$ has in its conscience a debt value ($T_{ix}$) for the service it received, but, prevented from the execution of the requested service, outsources it, which consists on the search of a third agent. This interactions generate exchange values, as shown in Figure 1:

- **Stage II:**
  1. $V_{xi}$: Credit value of agent $X$ for accomplishing a service of agent $I$.
  2. $T_{ix}$: Debt value of agent $I$ for the service received from agent agent $X$.
  3. $V_{terc(iy)}$: Outsourcing credit value from agent $I$ for $X$ in searching for a third agent. $I$ searches for an $Y$ agent, which has a credit value $V_{iy}$ with $I$, equivalent to $T_{ix}$.
  4. $T_{yi}$: Debt from agent $Y$ to agent $I$ for the satisfaction about the service received.
5. \( R_{yi} \): Investment value of agent \( Y \) for accomplishing a service for agent \( I \).

6. \( S_{iy} \): Satisfaction value of agent \( I \) for the service received from agent \( Y \).

7. \( S_{xi} \): Satisfaction of agent \( X \) for the service received from agent \( I \), generated by the investment of \( Y \) for \( I \). The satisfaction value is a function of agent \( X \)'s evaluation, and is represented by: \( S_{xi} = F_x(S_{iy}) \).

Formulations of Exchange Values in SEIA: Each obtained value in SEIA process is quantified based on the definitions and equations described in this section. Calculation of investment and satisfaction values were inspired in the measures theory LibQual\(^1\) [5], while credit and debit values were supported by [12] investigations.

**Definition 1** Investment values \( R \), Equation 1, represent an action or fact of service performance from one agent to other. It is a belief and their values are given by the ratio between investment adequacy \( (A_{inv}) \) and investment acceptance \( (Z_{a_{inv}}) \).

\[
R = \frac{A_{inv}}{Z_{a_{inv}}} \tag{1}
\]

Where:

1. **Investment Acceptance Zone** \( (Z_{a_{inv}}) \): It is the distance between the maximum investment value \( (v_{m_{inv}}) \) and expected investment value \( (v_{e_{inv}}) \). Thus: \( Z_{a_{inv}} = v_{m_{inv}} - v_{e_{inv}} \).
   (a) **Maximum investment value** \( (v_{m_{inv}}) \): It is an individual and differentiated belief of each agent. This value depends on the agent’s characteristics and represents everything that the agent can invest in the fulfillment of a service. It can be calculated as the weighted average of all of its values if there are more than one maximum.
   (b) **Investment expectation value** \( (v_{e_{inv}}) \): It is also a particular state of each agent, denotes the minimum possible investment that the agent plans to accomplish. As well as the maximum investment value, it can be calculated by the weighted average of expected minimums if there are more than one value.

2. **Investment adequation** \( (A_{inv}) \): Portrays the appropriate space for the agent’s investment conditions. It represents the distance between the maximum investment value \( v_{m_{inv}} \) and the observed investment value \( v_{o_{inv}} \), as follows: \( A_{inv} = v_{m_{inv}} - v_{o_{inv}} \).
   (a) **invested value** \( v_{o_{inv}} \): The invested value that the agent realizes while carrying out the required actions to achieve the service. The calculation of \( v_{o_{inv}} \) is represented by the investment factors weighted average \( f_1, f_2, f_3, ... , f_n \) and the weights \( (p_i) \): \( p_1, p_2, p_3, ... , p_n \). Each factor is a completed action and each weight is the importance factor that agent

\(^1\) Available at [http://www.libqual.org/](http://www.libqual.org/)
gives to that factor, obtaining: \( \psi_{invo} = (p_1i_1 \times f_1i_1) + (p_2i_2 \times f_2i_2) + \ldots + (p_ni_n \times f_ni_n)/(p_1 + p_2 + \ldots + p_n). \) Both factor values and factor weights are linked to particular characteristics of each agent, being different for each one.

When the agent’s observed investment level is less than the maximum investment value it can perform the investment is suitable. If the agent’s observed investment level is more than the maximum investment value, the investment is not suitable. In this case, the investment will not be made, because it overcomes the offer possibilities. Thus, the exchange will not be fulfilled. When the agent’s observed investment is less than the expected value, the investment is not significant, but the exchange process must continue.

**Definition 2**
The virtual credit value, \( V \), resultant from performed services, is the value that the agent finds itself worthy of return from the provision of a service it has performed. Depending of the scenario, the agent can valorize or depreciate its investment. This variation depends on the characteristics of each agent, which for the formulation is represented by an appreciation or depreciation rate of the amount invested, \( k_{inv} \). This rate assists in calculating the virtual value of credit that the agent finds itself worthy, as presented in Equations 2 e 3:

\[
\begin{align*}
\text{Valorization: } V &= R + (1 - R)k_{inv} \\
\text{Depreciation: } V &= (1 - k_{inv})R
\end{align*}
\]

**Definition 3**
The satisfaction value, \( S \), is the resulting pleasure of the performance of what was expected or desired from the received service. It is calculated as the average between the adequation of the service \( A_{serv} \) and the zone of acceptance of the service \( Z_{satisf} \), as specified in Equation 4.

\[
S = \frac{A_{satisf}}{Z_{satisf}}
\]

Where:

1. **Satisfaction acceptance zone** \( (Z_{satisf}) \): It is the space represented by the distance between the minimum satisfaction value, \( v_{min_{satisf}} \), and the desired satisfaction value, \( v_{des_{satisf}} \). This way: \( Z_{satisf} = v_{des_{satisf}} - v_{min_{satisf}} \). Every agent will accept what is above its minimum satisfaction value to its desired satisfaction value, generated after receiving a service.
   (a) **Desired satisfaction value** \( (v_{des_{satisf}}) \): It is an individual and differentiated belief for each agent. This value depends on qualities and characteristics of the agent. It represents the agent expectative about a service it is about to receive. It can be calculated as the weighted average of all of its maximum values if there are more than one.
   (b) **Minimum satisfaction value** \( (v_{min_{satisf}}) \): It is a particular condition in each agent and the value that makes the service acceptable. Any value below it causes the service to be considered unacceptable for the agent,
disabling the possibilities of the exchange process happen. It is possible to calculate the weighted average of the minimum values if there are more than one value.

2. Satisfaction adequation ($A_{satisf}$): It means that the satisfaction is appropriate to the prevailing conditions and circumstances. It is calculated as the difference between the observed value and the minimum value, as follows:

\[ A_{satisf} = v_{o,satisf} - v_{m,satisf} \]

(a) Observed satisfaction value ($v_{o,satisf}$): It is the satisfaction that the agent generates when a service is received. It is calculated as the weighted average of factors, $f_{s_1}$, $f_{s_2}$, ..., $f_{s_n}$, and weights, $p_{s_1}$, $p_{s_2}$, $p_{s_3}$, ..., $p_{s_n}$. Each factor is an action to evaluate and weights represent the degree of importance of each factor to the agent, as shown in:

\[ v_{o,satisf} = (p_{s_1} \times f_{s_1}) + (p_{s_2} \times f_{s_2}) + ... + (p_{s_n} \times f_{s_n}) / (p_{s_1} + p_{s_2} + ... + p_{s_n}) \]

Both the value of the factors and the weights are attached to particular characteristics of each agent, differing for each agent.

When the agent satisfaction level is above the minimum value, then the service is adequate. If the opposite occurs, the value of satisfaction obtained is below the minimum value, then the service, according to the agent, is not suitable. In this situation, the service will no longer be considered a service, and it will not carry out the exchange process between the agents. If the user-perceived value is higher than the desired value, then the service is considered of superior quality.

Definition 4 The virtual value of debt, $T$, generated as a result from the satisfaction about the services received, is the value that the agent thinks that is its debt. In certain scenarios, the agent can appreciate or depreciate the received value of satisfaction, influencing the final output value. This variation of satisfaction depends on the characteristics of each agent and is represented as a rate, $k_{satisf}$, of appreciation or depreciation of the value of satisfaction, as shown in equations 5 and 6:

\[ \begin{align*}
\text{Valorization} : & \quad T = S + (1 - S)k_{satisf} \\
\text{Depreciation} : & \quad T = (1 - k_{satisf})S
\end{align*} \]

Definition 5 The credit outsourcing value, $V_{terc(i|y)}$, is generated when there is a debt whose value cannot be paid by the agent itself. In this case, the agent will look for another agent with whom it has a credit value, which must be equivalent to the debt value it has with $X$, to request the performance of the requested service. This behavior is shown in Equation 7

\[ V_{terc(i|y)} = \begin{cases} V_{iy} & \text{If } (\exists y)(T_{ix} \simeq V_{iy}) \\ 0 & \text{If } (\forall y) \end{cases} \]

2.2 Dependence in SEIA

The agents $X$, $I$ and $Y$, which are interacting in services exchanges, produce exchange values, $(r, s, t, v)$, for each iteration, called exchanges values belief memory. In addition, each agent has a list of services it can offer. This generates
between the agents explicit and implicit dependencies, as presented in Figure 2. The explicit dependencies are created between agents who know each other and they had some type of exchange interaction in the provision of services, like $X - I$ and $I - Y$. The implicit dependencies are generated between agents who knows each other, $X - Y$, but they have done some interaction in the exchange of services through an intermediary agent, who interacted with both of them.

Fig. 2: Dependence Relations in SEIA

In addition, may be different degrees of dependency between the different agents: **Very weak dependencies**, characterized by an agent who has no credit to claim. It needs an agent to perform a service that it could make, since it can do the job. **Weak dependencies**, characterized by an agent who has a credit to charge and needs to perform a service that it can accomplish, as it has the knowledge to perform the service. **Strong dependencies** are characterized by an agent who does not have a credit to charge and needs to perform a service that it cannot accomplish by itself, depending of the other agent knowledge. Finally, **very strong dependencies**, are those where the agent have a credit to earn and it needs a service it does not have the knowledge to perform.

The very weak dependencies and the weak dependencies have the particularity of ceasing to be dependencies, because independent of whether or not there is a credit charge, the agent itself can perform the service, if necessary, to achieve its goal, but decided to ask another agent, based on certain circumstances. Note that the dependence degree varies and that the higher the credit value, the stronger is the dependence and the lower is the power of the agent to perform the service, the stronger is the dependence it has. The degree of dependence is calculated taking into account the dependence definition 6.

**Definition 6** The degree of dependence represents numerically how much an agent depends on the other. This calculation takes in account the credit value $V$, and the capacity of performing the service value $W$. The capacity of performing the service value varies between 0 and 1. Zero (0) represents that the agent is unable to do the service and need another agent for the service to be performed. One (1) represents that agent has all the conditions to perform the service it needs, so it is up to it to decide whether to outsource the service of not. Values between 0 and 1 indicate different levels of the capacity of performing the service. Equation 8 represents the degree of dependence.

\[
G_{dep} = V + (1 - V)(1 - W)
\]
2.3 Reputation in SEIA

The reputation model in SEIA, as shown in Figure 3, consists of three agents \(X, I\) and \(Y\), and it is characterized by the fact that each agent is making direct or indirect evaluation of the services it receives in each interaction of the SEIA. Thus, the reputation of an agent depends on its exchange values history and also on the knowledge of the existing dependence degree.

![Fig. 3: Reputation in SEIA](image)

**Construction of Beliefs in SEIA**: The construction of beliefs about the reputation value of exchange values, \(V_{rep}\), is a product of the comparison between the observed satisfaction value desired satisfaction value, in the provision and contra-provision of services in the first and second stages of the exchange, respectively, as detailed below:

- \(I\) evaluates the service performed by agent \(X\) in Stage I - Exchange 1:
  - \(I\) generates a satisfaction value about the service received, product of the investment made by agent \(X\).
  - \(I\) analyses \(X\) by comparison in the trichotomy of the observed satisfaction value in the reception of the service and the desired satisfaction, as follows: \(S_{ix} > v_d\text{satisf}\); \(S_{ix} = v_d\text{satisf}\) and \(S_{ix} < v_d\text{satisf}\), to obtain a reputation value through the exchange values \(V_{Rep} \in \{1; 0.5, 0\}\), respectively.
  - \(I\) makes a record about the reputation values obtained from the trichotomy about the exchange it made with \(X\).

- \(Y\) evaluates the service made by agent \(I\) in Stage I - Exchange 2
  - \(Y\) generates a satisfaction value about the service received, product of the investment made by agent \(I\).
  - \(Y\) analyses \(I\) by comparison in the trichotomy of the observed satisfaction value in the reception of the service and the desired satisfaction, as follows: \(S_{yi} > v_d\text{satisf}\); \(S_{yi} = v_d\text{satisf}\) and \(S_{yi} < v_d\text{satisf}\) to obtain a reputation value through the exchange values \(V_{Rep} \in \{1; 0.5, 0\} rbrace\), respectively.
  - \(Y\) makes a record about the reputation values obtained from the trichotomy about the exchange it made with \(I\).

- \(I\) evaluates the service delivered by agent \(Y\) in Stage II:
  - \(I\) generates a satisfaction value in the reception of the service, product of the investment made by agent \(Y\).
• I analyses Y by comparison in the trichotomy of the observed satisfaction value in the reception of the service and the desired satisfaction, as follows: $S_{iy} > vd_{satisf}$; $S_{iy} = vd_{satisf}$ and $S_{iy} < vd_{satisf}$, to obtain a reputation value through the exchange values $V_{Rep} \in \{1; 0,5; 0\}$, respectively.

• I makes a record about the reputation values obtained from the trichotomy about the exchange it made with Y.

- X evaluates the service performed by agent I in Stage II:
  • X generates a satisfaction value about the service received, product of the investment made by agent I.
  • X analyses I by comparison in the trichotomy of the observed satisfaction value in the reception of the service and the desired satisfaction, as follows: $S_{xi} > vd_{satisf}$, $S_{xi} = vd_{satisf}$ and $S_{xi} < vd_{satisf}$, to obtain a reputation value through the exchange values $V_{Rep} \in \{1; 0,5; 0\}$, respectively.
  • X makes a record about the reputation values obtained from the trichotomy about the exchange it made with I.

Direct and indirect evaluations are represented as direct reputations and indirect reputations, respectively. So, direct reputation comes from agents who know each other, and indirect reputation comes from agents that do not know each other directly, but exchanged a service indirectly. Figure 3 shows how reputation is distributed in a SEIA. Agent X generates a direct reputation ($Rep_d$) about agent I, and an indirect reputation ($Rep_i$) about agent Y. Agent I generates direct reputations, ($Rep_d$), about agents X and Y. Finally, agent Y generates a direct reputation, ($Rep_d$), about agent I.

Besides the reputation value from exchange values, $V_{Rep}$, the degree of dependence, $G_{dep}$, is also part of the calculation of the reputation of an agent, as stated in the Definition 7

Definition 7 Each agent X, I, Y will have a neutral initial reputation specified for each system. The reputation value, $Rep$, defined below, is the balance of the evaluation that the agent will get from other agents. $Rep = V_{Rep} + (1 + V_{Rep}) \times G_{dep} \times G_{dep} \geq 0,5 \land V_{Rep} \in \{1; 0,5\}; (1 - G_{dep}) \times V_{Rep} \times G_{dep} \geq 0,5 \land V_{Rep} = 0; V_{Rep} \times G_{dep} \times G_{dep} < 0,5 \land V_{Rep} \in \{1; 0,5; 0\}$

Finally, the verbal equivalent of the reputation value will belong to the set of positive real numbers ranging from 0 to 1, considering the reputation as bad when it is in range $[0;0,25]$, that is $Rep \in [0;0,25]$; Bad $\iff Rep \in [0,25;0,5]$; Regular $\iff Rep \in [0,5;0,75]$; Good $\iff Rep \in [0,75;1]$; and Excellent $\iff Rep = 1$

3 Trust Transfer in SEIA

The trust transfer process in SEIA, as modeled in Figure 4, consists of three agents X, I and Y. Agent X, who does not know agent Y, will evaluate the possibility to transfer its trust to an agent with whom it had no direct iterations, but has information about.
Agent X about Agent I:
1. X has a potential goal ($V_{xi}$) to reach in certain circumstances ($T_{ix}$). X believes that, if it wants to reach that goal and circumstances do not change, then:
   a) I will be able to make an action $R_{ix}$
   b) I, performing the action $R_{ix}$ ensures to reach the goal $V_{xi}$, obtaining a satisfaction value $S_{xi}$ desired by X
   c) I has the intention of performing $R_{ix}$ because it has a debt with X

Agent I about Agent Y: Since I cannot perform the service requested by X, it searches for a third agent, Y, who is able to do it. The same that happened between X and I will happen again between I and Y:
1. I has a potential goal $V_{terc(xiy)}$ to reach in certain circumstances ($T_{yi}$). I believes that, if it wants to reach its goal of paying its debt with X and the circumstances do not change, then:
   a) Y will be able to perform an action $R_{yi}$
   b) Y, performing the action $R_{yi}$, ensures to reach the I goal: $V_{terc(xiy)}$, and, that way, through I, X’s objective: $V_{xi}$. This provides the satisfaction values $S_{iy}$, desired by I, and $S_{xi}$, desired by X. Item [c)] Y has a predisposition to perform $R_{yi}$, because it has a debt with I.

Agent X about Agent Y:
Based on the trust transfer, agent X will believe in Y after getting the payment of the credit requested from agent I and embodied by Y. It can be said that:
1. If X has a potential goal ($V$) to achieve in certain circumstances ($T$), it believes that if it wants to achieve that goal and the circumstances do not change, then:
   a) Y will be able to perform an action $R$.
   b) Y, by performing the action $R$, ensures to reach X goal, represented by the credit value, $V$, that it wants to obtain, generating also the desired satisfaction value $S$.
   c) Y has the intention to make an investment, $R$, as part of a social exchange process.

This requires developing an analysis of the group’s beliefs. These beliefs are composed by the reputation of each agent, whereas the reputation value used is calculated taking into account the historical changes of values and relations of dependency, as described in the definition 7 about building reputation beliefs in SEIA.
**Definition 8** In a trust transference process, an $X$ agent can transfer his trust to a $Y$ agent that he never met if, and only if, there is an $I$ agent, known by $X$ and $Y$. Besides, for the $X$ agent, the final reputation of $I$, which provides information about $Y$’s reputation, must be greater of equal to $R_{acep}$, which is the acceptable reputation value set by $X$. After the fulfillment of this condition, $X$ evaluates the final reputation of $Y$ set by $I$, verifying again if it is greater or equal to $X$ acceptable reputation, as shown in equation (9).

$$If: \left[ \left( \sum_{i=1}^{n} \text{Rep}_{I1} \lor \text{Rep}_{I2} \lor \ldots \lor \text{Rep}_{In} \geq R_{acep} \right) \land \left( \sum_{i=1}^{n} \text{Rep}_{I1} \lor \text{Rep}_{I2} \lor \ldots \lor \text{Rep}_{In} \geq R_{acep} \right) \right] \Rightarrow V_{x \rightarrow y}^{transf}$$

(9)

Where:

- $V_{x \rightarrow y}^{transf}$: Trust transfer from $X$, with respect to $Y$.
- $n$: Total number of agents that provide information.
- $\text{Rep}_{I1} \lor \text{Rep}_{I2} \lor \ldots \lor \text{Rep}_{In}$: Final reputation values of the agents $I_1$ and $I_2$ and ... and $I_n$, which know agent $Y$ and interacted with agent $X$.
- $\text{Rep}_{I1}^{y} \lor \text{Rep}_{I2}^{y} \lor \ldots \lor \text{Rep}_{In}^{y}$: Final reputation values provided by each of the agents $I_1$ or $I_2$ or ... or $I_n$, about agent $Y$.
- $R_{acep}$: Acceptable reputation value set by agent $X$.

It is important to notice that the acceptable reputation value set by $x$ is a function of its intrinsic characteristics.

**For an interaction in triads of agents**: $X$, $I$, $Y$:

1. $X$ wants to interact with unknown agent $Y$ to reach its target in certain circumstances:
   - $X$ verifies the reputation value provided by $I$ about $Y$: $\text{Rep}_{I}^{y}$
2. $X$ transfers its trust to $Y$, by trusting in $I$ to reach its target when:
   - $X$, to rely on $I$, checks and compares if the reputation it has about $\text{Rep}_{I}^{y}$ is greater than or equal to the reputation value that it considers acceptable, as follows: $\text{Rep}_{I}^{y} \geq R_{acep}$
3. $X$, do not transfer its trust to $Y$, not trusting in $I$, when:
   - $X$ checks and compares if the reputation it has about $I$, $\text{Rep}_{I}^{x}$ is under the reputation value that it considers acceptable, $\text{Acep}$: $\text{Rep}_{I}^{x} < R_{acep}$

Briefly, $X$ will trust that $Y$ would help it achieve its goal only if it is necessary and the existing information about $Y$ is favorable to its needs. Thus, in a future exchange, $X$ can transfer its trust and relate directly with $Y$.

4 **Final Considerations**

In the real world, it is very common to find intermediaries in exchange processes. Therefore, it is important to consider the extension of Piaget’s theory
about Social Exchanges to three agents. The presented models are a beginning
for the analysis of non-economic aspects of social exchange processes, and to
understand the behavior of agents in interactions with more than three agents,
taking into account the concepts of dependence, reputation and trust, inter-
dependently. The integration of the concepts of exchange values, dependence,
reputation and trust in the social exchange process in triads of agents enabled
to model an internal structure for each agent that interacts in these service
exchanges. The information of the exchange values generated by the evaluation
of the services will influence the reputation and the existing dependency rela-
tionships. This will guide the decision of the agents on the choice of partners
with whom they never had direct interaction. Thus, it becomes the focus for the
analysis of trust transfer process between agents. The proposed models can be
submitted to simulations and evaluations through case studies, to validate them
quantitatively.

References
1. D. B. Bromley, J. Wiley, and Sons. Reputation, image, and impression manage-
2. C. Castelfranchi and R. Falcone. Principles of trust for mas: Cognitive anatomy,
social importance and quantification. In International Conference of Multi-agent
3. C. Castelfranchi, R. Falcone, B. Firozabadi, and Y. Tan. Special issue on trust,
deception and fraud in agent societies. Applied Artificial Intelligence Journal, pages
based on Piaget’s theory of social exchanges using a BDI-fuzzy agent model. In 11th CBIC,
5. E. Herrera-Viedma, J. López-Gijón, and B. Avila. Satisfacción de usuarios y cali-
dad: El modelo libqual y su aplicación en bibliotecas españolas e iberoamericanas.
6. A. Herzig, E. Lorini, J. F. Hübner, and L. Vercouter. A logic of trust and reputa-
tion. Logic Journal of the IGPL, Normative Multiagent Systems, 18(1):214–244,
2010.
7. S. Marsh. Formalising Trust as a Computational Concept. Thesis, Department of
8. D. Pereira, L. Gonçalves, G. P. Dimuro, and A. R. C. Costa. Towards the self-
regulation of personality-based social exchange processes in multiagent systems.
11. J. S. Sichman and Y. Demazeau. A first attempt to use dependence situations as
a decision criterion for choosing partners in multi-agent systems. In Workshop on
12. A. Von Laer. Autorregulação de processos de trocas sociais em sma: um modelo de
sociedade de agentes bdi evolucionários e culturais no contexto do jacamo. Master,
York, NY, USA, 2002.
A Multiagent-Based Domain Transportation Approach for Optimal Resource Allocation in Emergency Management

Jihang Zhang, Minjie Zhang, Fenghui Ren and Jiakun Liu

Abstract In metropolitan regions, emergency events request urgent response within a short time limit in order to minimise the damage and the number of fatality. Most of these events require different resources that are usually distributed over a large area. How to efficiently allocate the distributed resources to an event is a challenging research issue. Traditional centralised resource allocation approaches have difficulties to find out the best resource assignment within the event’s time limits by considering the dynamics of the metropolitan environment and the event itself. In this paper, a multiagent-based decentralised resource allocation approach using domain transportation theory is proposed to handle an emergency event with multiple tasks. Experimental results indicates that the proposed approach can effectively generate the optimal resource allocation plans by considering multiple factors of an emergency event.

1 Introduction

In recent decades, with the rapid increase of population in metropolitan regions, public emergency departments suffer significant pressure about how to efficiently and effectively allocate rescue resources for emergency events (i.e. vehicle accident, fire, terrorist attacks, etc). Generally, these emergency events have the three common characteristics, including that (1) they are hard to be predicted in dynamic environments such as metropolitan regions; (2) they usually require multiple different resources with different usage costs, mobilities, availabilities, ownerships and functionalities (i.e. a vehicle acci-
dent might require ambulances, polices and fire fighters); and (3) they have strict time limits for emergency departments to response and allocate rescue resources. Currently, most resource allocation processes for the emergency events in metropolitan regions are still operated manually, which is highly inefficient. This is because in a metropolitan region, large numbers of rescue resources with different functionalities and availabilities are distributed over an extensive area. Emergency event operators usually have difficulties to efficiently find out the optimal resource allocation for an emergency event due to a large number of possibilities.

In recent years, agent and multi-agent systems have been becoming promising technologies for resource allocation in emergency situations and many useful approaches have been proposed to assist human operators to make decisions [1–3,5,6]. Aygul et al. [3] proposed an agent-based solution for finding the optimal allocation of emergency medical resources, which was implemented based on service-oriented architecture. Since their approach requires the global information of all resources, so it might not available in most real life situations. Beatriz et al. [5] proposed a multi-agent system to allocate ambulances for emergency medical events by using the contract net protocol and a winner determination algorithm to find out the optimal ambulances allocation. However, their approach is restricted to handle ambulances, so it might not suitable for emergency events with the requirements of different resources. Fiedrich [1] proposed an multi-agent system for optimal resource allocation for large natural disasters based on High Level Architecture (HLA), which was capable of sequentially allocating appropriate resources to tasks. However, sequential resource allocation might increase the number of fatality of tasks in the later position of the task queue.

In order to overcome the above limitations, this paper proposes an agent-based decentralised resource allocation approach for an emergency event in metropolitan regions. The proposed approach first converts the resource allocation problem of a single emergency event into different resource allocation tasks. Then, different agents propose their resource allocation proposals to these tasks simultaneously. Finally, for each task, the domain transportation theory is used to combines these proposals and find out the optimal resource allocation that minimises the total allocation cost. The major contributions of the proposed approach are that (1) the proposed approach is designed to efficiently allocate distributed resources in decentralised manner, which is more applicable and practical comparing with centralised approaches; (2) the proposed approach is designed to effectively handle different types of emergence events that require resources with different functionalities; (3) the proposed resource allocation framework can simultaneously allocate different resources to different tasks of an emergency event; and (4) the proposed resource allocation algorithm is highly adaptable and flexible, which is capable of generating optimal allocation solutions for emergency situations in different domains with different cost functions or attributes.
The rest of this paper is organised as follow. Section 2 gives problem description and definitions. Section 3 introduces the theoretical foundation of the optimal resource allocation. Section 4 describes the agent-based solution for the implementation of the proposed resource allocation approach. Section 5 demonstrates the experimental results and provides analysis. Section 6 gives the conclusion and outlines the future work.

2 Problem Description and Definitions

This section introduces the important definitions that are used in the proposed resource allocation approach and the fundamental problem that the proposed approach is trying to address.

2.1 Definitions of Domain Knowledge

Definition 1 (Environment). An environment is represented by a city map, which is defined as an undirected graph, \( G = [V, E] \), where \( V = \{v_1, v_2, ..., v_i\} \) is a set of nodes, which represent important locations in a metropolitan region, and \( E = \{e_1, e_2, ..., e_j\} \) is a set of edges, which represent the paths between the nodes. \( e_j \) is further defined as a two-tuple, \( e_j = (v_o, v_p) \), where \( v_o, v_p \in V \) are the nodes that be connected by \( e_j \).

Definition 2 (Resource). A resource is defined as a seven-tuple, \( res = (rty, ser, fun, rlo, ava, vel, exp) \), where \( rty \in \{\text{facility, mobile}\} \) represents the type of resources, where facility refers to unmovable rescue resources, such as fire stations, hospitals, etc, and mobile refers to rescue vehicles and personnel; \( ser \in \{\text{fire \\& rescue, medical, police}\} \) represents the type of emergency services that \( res \) can provide; \( fun \in \{\text{fire fighter, fire truck, police officer, police car, ambulance, medical personnel}\} \) represents \( res \)'s functionality; \( rlo \in V \) represents \( res \)'s current location; \( ava = \{0, 1\} \) represents \( res \)'s availability, where 0 indicates unavailable and 1 indicates available; \( vel \in (0, +\infty) \) represents \( res \)'s average velocity in kilometre per hour (km/h) and \( exp \) represents \( res \)'s money expenditure in per hour when \( rty = \text{facility} \), while \( exp \) represents \( res \)'s money expenditure in per hour, per kilometre when \( rty = \text{mobile} \).

In the proposed approach, set \( \text{REE} \) indicates all resources in \( G \). Besides, it is assumed that the money expenditure \( exp \) of \( res \) is known by local emergency departments. Furthermore, the set of resource services and functionalities defined above could be extended in real-world applications.

Definition 3 (Task). A task is defined as a three-tuple, \( tas = (dea, ser, TR) \), where \( dea \) represents the deadline for resources to be allocated to \( tas \); \( ser \) represents the type of emergency service that \( tas \) requires to make response and \( TR = \{tr_1, tr_2, ..., tr_e\} \) represents a set of required resources for completing \( tas \). \( tr_e \) is further defined as a two-tuple, \( tr_e = \{rty, fun\} \).

In the proposed approach, it is assumed that local emergency departments have the knowledge to estimated a task’s deadline \( dea \) based on an event’s severity.
**Definition 4 (Event).** An event is defined as a five-tuple, \( \text{eve} = (\text{con}, \text{SER}, \text{elo}, \text{sev}, \text{TAS}) \), where \( \text{con} \in \{\text{fire, rescue, loss of life, damage to health, security of person, security of property}\} \) represents \( \text{eve} \)'s content; \( \text{SER} \subseteq \{\text{fire & rescue, medical, police}\} \) represents a set of emergency services required by \( \text{eve} \); \( \text{elo} \in \mathbb{V} \) represents \( \text{eve} \)'s location; \( \text{sev} \in [1, 10] \) represents \( \text{eve} \)'s severity and \( \text{TAS} = \{\text{tas}_1, \text{tas}_2, ..., \text{tas}_k\} \) represents a sequence of tasks that need to be completed for \( \text{eve} \).

**Definition 5 (Resource Allocation Proposal).** A resource allocation proposal for an event is defined as a two-tuple, \( \text{rap} = (\text{eve}, \text{RES}) \), where \( \text{RES} \subseteq \text{REE} \) represents a set of resources that be proposed for completing tasks in \( \text{eve} \). Besides, a resource allocation proposal for a single task in the event is defined as a two-tuple, \( \text{rap}_k = (\text{tas}_k, \text{RES}) \), where \( \text{rap}_k.\text{RES} \subseteq \text{rap.RES} \).

In the proposed approach, the cost of resource allocation is calculated by cost functions. Usually, different emergency events might need to use different cost functions, which might involve different cost attributes. At here, a cost function is defined, by considering two significant factors in emergency resource allocation, i.e., money expenditure and time. In the following, the cost for allocating a single resource to a single task is firstly defined.

**Definition 6 (Cost Function).** A cost function for a single resource’s allocation is defined by Equation 1:

\[
\text{COR}(\text{eve.task}_k, \text{res}) = \begin{cases} 
\text{res.exp, if } \text{res.rty} = \text{facility} \\
\text{res.exp} \times \text{DIS}(\text{res.rlo, eve.elo}) \times \frac{\text{DIS}(\text{res.rlo, eve.elo})}{\text{res.vel}} \times \text{DLINE}(\text{eve.task}_k, \text{res}), \text{if } \text{res.rty} = \text{mobile}, 
\end{cases}
\]

where \( \text{DIS}(\text{res.rlo, eve.elo}) \) is a function that return the distance of a passable road between resource location \( \text{res.rlo} \) and event location \( \text{eve.elo} \), which could be implemented by various path searching algorithms, such as Dijkstra’s and A* algorithms [7, 9]. \( \text{DLINE}(\text{eve.task}_k, \text{res}) \) is a function that is used to determine whether \( \text{res} \) can be allocated within \( \text{eve.task}_k \)'s deadline, which is further defined by Equation 2:

\[
\text{DLINE}(\text{eve.task}_k, \text{res}) = \begin{cases} 
1, \text{if } \frac{\text{DIS}(\text{res.rlo, eve.elo})}{\text{res.vel}} \leq \text{eve.task}_k.\text{dea} \\
+\infty, \text{if } \frac{\text{DIS}(\text{res.rlo, eve.elo})}{\text{res.vel}} > \text{eve.task}_k.\text{dea}
\end{cases}
\]

Based on above terms, the cost function for allocating all required resources to a single task is defined by Equation 3:

\[
\text{COT}(\text{eve.task}_k, \text{rap}_k) = \sum_{\text{res} \in \text{rap}_k.\text{RES}} \text{COR}(\text{eve.task}_k, \text{res})
\]

Furthermore, the cost function for allocating all required resources to all tasks in a single event is defined by Equation 4:
\[ COE(eve, rap) = \sum_{eve.tas_k \in eve.TAS} COT(eve.tas_k, rap_k) \]  

**2.2 Problem Description**

For an emergency event \( eve \), there could be different resource allocation proposals. In the proposed approach, the all possible resource allocation proposals for \( eve \) are represented as set \( RAP \). The main objective of the proposed approach is to search an optimal resource allocation proposal \( rap^* \in RAP \) for \( eve \). The objective function for an event’s resource allocation is formally defined by Equation 5:

\[
OBJE = \arg \min_{rap^* \in RAP} COE(eve, rap^*) \text{ subject to } rap^* \in \text{REE}
\]  

where \( OBJE \) represents the name of objective function for an event’s resource allocation and \( rap^* \in \text{REE} \) means that the proposed resources in \( rap^* \) must belong to the available resources in the environment G. The event objective function indicates that the optimal resource allocation proposal \( rap^* \) must have the minimal allocation cost in \( RAP \). Besides, the proposed approach assumes there is always enough resources in \( \text{REE} \) to be allocated for \( eve \).

However, due to the fact that an emergency event usually require resources with different types and functionalities, searching a complete \( rap^* \) could be a complicated and time-consuming process. In order to efficiently solve this searching problem, the proposed approach creates a set of resource allocation tasks \( eve.TAS \) for \( eve \). For each task \( tas_k \) in \( eve.TAS \), it only requires resources that provide the same type of emergency service (i.e. \( res.ser \)). By doing so, the searching of \( rap^* \) for \( eve \) is converted to the concurrent searching of \( rap^*_k \in RAP_k \) for each task \( tas_k \) in \( eve.TAS \), where \( RAP_k \) represents the all possible or available resource allocation proposals for \( tas_k \). The objective function of resource allocation for a task \( tas_k \) is formally defined by Equation 6:

\[
OBJT = \arg \min_{rap^*_k \in RAP_k} COT(eve.tas_k, rap^*_k) \text{ subject to } rap^*_k \in \text{REE}
\]

where \( OBJT \) represents the name of the objective function for an task’s resource allocation.

**3 Theoretical Foundation of the Domain Transportation for the Optimal Resource Allocation**

Domain transportation theory is a linear programming method to minimise the cost of relocating resources [4]. In domain transportation, the resource allocation problem of a task can be described as a resource mapping problem from the available resources in an environment to the required resources of the task, which is formally represented by Equation 7:
\[ \text{rap}_k : \text{REE} \rightarrow \text{tas}_k, \text{TR} \]

 Apparently, there are many different mapping proposals (i.e. \( \text{RAP}_k \)) from domain \( \text{REE} \) to domain \( \text{tas}_k, \text{TR} \). In this paper, domain transportation theory is used to find out \( \text{rap}^*_k \) for \( \text{tas}_k \), which can fulfil Equation 6.

More precisely, let \( \text{REE}_e(y_i) \) denote the amount of functionality \( e \) resource at location \( y_i \in V \). Let \( x_k = \text{eve.tas}_k \) and \( \text{tr}_e(x_k) \) represent the required amount of functionality \( e \) resource at task \( x_k \). The cost of transferring the resource at \( y_j \) to the task \( x_k \) is given by Equation 1, i.e. \( \text{COR}(x_k, y_j) \).

An admissible allocation proposal \( \text{rap}_k \) is a mapping from \( V \) to \( \text{TAS} \) satisfying the balance condition that for any subset \( E \subset \text{TAS} \),

\[ \sum_{x_k \in E} \text{tr}_e(x_k) = \sum_{y_j \in \text{rap}^{-1}_k(E)} \text{REE}_e(y_j), \quad (8) \]

where \( \text{rap}^{-1}_k \) is the inverse mapping of \( \text{rap}_k \). As before, let \( \text{RAP}_k \) denote the set of all admissible allocation proposals. The purpose of Equation 6 is to find an optimal proposal \( \text{rap}^*_k \in \text{RAP}_k \) such that

\[ \text{COT}(x_k, \text{rap}^*_k) = \min_{\text{rap}_k \in \text{RAP}_k} \text{COT}(x_k, \text{rap}_k). \quad (9) \]

From optimal transport theory, Equation 9 can be transferred to the following linear programming

\[
\max \left\{ \sum_{x_k \in \text{TAS}, y_j \in V} u(x_k)\text{tr}_e(x_k) + v(y_j)\text{REE}_e(y_j) : u(x_k) + v(y_j) \leq \text{COR}(x_k, y_j) \right\}.
\]

Moreover, there exists a maximiser \((u^*, v^*)\) (unique up to a constant) achieving the above maximum. \( u^* \) is called a potential, and \( v^* \) is its dual potential. The pair \((u^*, v^*)\) also satisfies a generalised Legendre duality associated with the cost function \( \text{COR} \). Hence, for each \( x_k \), there exists a unique \( y_i \) such that the equality holds in the constraint, namely \( u^*(x_k) + v^*(y_i) = \text{COR}(x_k, y_i) \), which means the task \( x_k \) requires the resource from location \( y_i \). Thus, one can construct a mapping \( \text{inv}^*_k : x_k \in \text{TAS} \rightarrow y_j \in V \). From optimal transport theory, by differentiating the above equation one can see that

\[
\nabla u^*(x_k) = \nabla_x \text{COR}(x_k, \text{inv}^*_k(x_k)), \\
y_i = \text{inv}^*_k(x_k) = \nabla^{-1}_x \text{COR}(x_k, \nabla u^*(x_k)). \quad (11)
\]

In fact, one can further show that \( \text{inv}^*_k \) is exactly the inverse of the optimal allocation proposal \( \text{rap}^*_k \).

Therefore, to construct \( \text{rap}^*_k \), it suffices to follow the following steps:

1. From the given data \( \{\text{tr}_e, \text{TAS}, \text{REE}_k, V\} \), formulate the linear programming Equation 10;
2. Solve Equation 10 to find out a potential \( u^* \);
Using Equation 11 to construct the mapping $inv^*_k : x_k \in TAS \rightarrow V$, which implies that task $x_k$ requires the resource from the area $REQ := inv^*_k(x_k) \subset V$.

4. Take the inverse, we obtain the optimal allocation proposal $rap^*_k : \text{REQ} \rightarrow x_k$, which can inform the agent how to distribute the resource in the optimal way.

4 Agent-based Decentralised Resource Allocation

The proposed resource allocation approach is implemented based on agent and multi-agent technologies, due to agents ability of autonomous reasoning, intelligent decision making, group coordination and collaboration [8]. This section gives detail description of agents’ definitions, resource allocation framework and process.

4.1 Definitions of Agents

Generally, there are four types of agents in the proposed resource allocation approach, which are response agent, mobile agent, facility agent and deployment agent. Each agent’s definition is described as follows.

Definition 7 (Response Agent). A response agent is represented by $ra$, which has the information of a specific emergency event. A response agent has four major functionalities, which are (1) identifying event content $\text{eve.con}$ for a new event $\text{eve}$; (2) identifying the emergency services $\text{eve.SER}$ that is required by $\text{eve}$ based on $\text{eve.con}$; (3) identifying a set of tasks $\text{eve.TAS}$ for $\text{eve}$ based on $\text{eve.ser}$; and (4) sending $\text{eve.TAS}$ to a deployment agent.

Definition 8 (Mobile Agent). A mobile agent is represented by $ma$, which has the information of a specific mobile resource $ma.res$. A mobile agent has two major functionalities, which are (1) managing a mobile resource $ma.res$; and (2) implementing resource allocation paths after receiving resources allocation commands.

Definition 9 (Facility Agent). A facility agent is represented by $fa$, which has the information of a specific facility resource $fa.res$ and a set of mobile agents. More precisely, $fa.MA = \{ma_1, ma_2, ..., ma_j\}$ represents a set of mobile agents that belong to $fa$ and $\text{REF} = \{fa.res\} \cup \{ma_1.res, ma_2.res, ..., ma_j.res | ma_j \in fa.MA\}$ represents all resources under $fa$’s management. A facility agent has three major functionalities, which are (1) managing a facility resource $fa.res$; (2) generating resource allocation proposals for tasks based on $fa.REF$; and (3) informing its mobile agents to execute resources allocation commands after receiving the commands from a deployment agent.

Definition 10 (Deployment Agent). A deployment agent is represented by $da$, which has the information of a specific emergency event. A deployment agent has three major functionalities, which are (1) informing an event’s tasks information (i.e. $\text{eve.TAS}$) to facility agents that are located in its circle communication area, represented by $da.com$; (2) combining and generating
the optimal resource allocation proposal \( \mathit{rap}^*_k \) for a task based on a set of proposals (i.e. \( \mathit{RAP}_k \)) submitted by facility agents; and (3) informing relevant facility agents to execute \( \mathit{rap}^*_k \).

### 4.2 Resource Allocation Framework and Process

The proposed resource allocation approach is implemented by a multi-agent system (MAS), which includes a task identification module, resource identification module, proposal generation module, optimal allocation module and proposal execution module. The framework of the MAS is depicted in Fig. 1.

As depicted in Fig. 1, an event’s resource allocation process involves one response agent, one deployment agent, multiple mobile agents and facility agents. The allocation process is formally described by Algorithm 1.

#### Algorithm 1 : Resource Allocation Process

**Input:** \( \mathit{eve} \)

1. assign \( \mathit{ra} \) to \( \mathit{eve} \)
2. \( \mathit{ra} \) identifies \( \mathit{eve}.\mathit{con} \)
3. \( \mathit{ra} \) identifies \( \mathit{eve}.\mathit{SER} \) based on \( \mathit{eve}.\mathit{con} \)
4. \( \mathit{ra} \) identifies \( \mathit{eve}.\mathit{TAS} \) based on \( \mathit{eve}.\mathit{SER} \)
5. \( \mathit{ra} \) sends \( \mathit{eve}.\mathit{TAS} \) to \( \mathit{da} \)
6. \( \mathit{da} \) calculates circle communication area \( \mathit{da}.\mathit{com} \)
7. \( \mathit{da} \) locates \( \mathit{FA} \) in \( \mathit{da}.\mathit{com} \)
8. for all \( \mathit{tas}_k \in \mathit{eve}.\mathit{TAS} \) do
9. \( \mathit{da} \) creates \( \mathit{RAP}_k \) and \( \mathit{FA}_k \)
for all \( f_a_i \in FA \) do

if \( f_a_i.res.ser = task.ser \) then

da updates \( FA_k = \{ f_a_i \} \cup FA_k \)
da sends \( task \) to \( f_a_i \)
f_a_i finds \( rap_i^k : f_a_i.REF \rightarrow task.TR \)

if \( tcu \leq PDLINE(task.dea, eve.sev) \) then

\( f_a_i \) submits \( rap_i^k \) to \( da \)
da updates \( RED_k = \{ rap_i \} \cup RED_k \)
da sends \( task \) to \( f_a_i \)

end for
end for

while \( |eve.TAS| > 0 \) do

for all \( task \in eve.TAS \) do

if \( tcu \geq PDLINE(task.dea, eve.sev) \lor \forall f_a_g \in FA_k : f_a_g \) submit \( rap_g^k \) then

if \( RAP_k \) contains enough resources for \( task \) then

da expend \( da.com \) by double

process goes back to Line 7

else

da finds \( rap_i^* : RED_k \rightarrow task.TR \)
da updates \( eve.TAS = eve.TAS \setminus \{ task \} \)
da informs agents to execute \( rap_i^* \)

end for

end for

end while

The resource allocation process shown in Algorithm 1 includes six steps, which are explained as follows.

**Step 1: (Lines 1-5)** When an emergency event \( eve \) happened, a new response agent \( ra \) is assigned to \( eve \) to identify the emergency content \( eve.con \). Then, \( ra \) needs to identify the emergency services \( eve.SER \) required by \( eve \) according to \( eve.con \). For example, when \( eve.con = fire \), the required emergency services could be \( eve.SER = \{ fire & rescue, medical, police \} \). After emergency service identification, \( ra \) needs to acquire the information of the resources required by each emergency service, which may provided by human operators or other external agents. Then, \( ra \) converts each of emergency service to a task. Finally, \( ra \) sends \( eve.TAS \) to a deployment agent \( da \).

**Step 2: (Lines 6-13)** After receiving \( eve.TAS \), \( da \) first needs to calculate an communication area \( da.com \), which is a circle centred at the event’s location \( eve.elo \) and measured by square kilometres. \( da.com \) is calculated by Equation 12:

\[
da.com = \pi \times (avv \times (\frac{\sum_{task \in eve.TAS} task.dea}{|eve.TAS|} - tcu))^2
\]

where \( avv \) represents the average moving velocity (km/h) of all required mobile resources in \( eve.TAS \) and \( tcu \) represents the current time. In the proposed approach, it is assumed that local emergency departments have the knowledge of the average velocity of different functionalities’ resources.

After the calculation of \( da.com \), \( da \) needs to locate all facility agents inside \( da.com \), which is represented by set \( FA = \{ f_a_1, f_a_2, ..., f_a_i \} \) (Line 9). Then, \( da \) sends each \( task \) to relevant facility agents in \( FA \) based on \( task \)'s emergency
service requirement $tas_k\.ser$. At the same time, $da$ also creates a facility agent contact list $FA_k = \{fa_1, fa_2, ..., fa_g\}$ ($FA_k \subseteq FA$) and resource allocation proposal list $RAP_k = \{rap_{1k}, rap_{2k}, ..., rap_{gk}\}$ for each $tas_k$ in eve.TAS.

**Step 3:** (Lines 14-17) After a facility agent $fa_i \in FA$ receives $tas_k$, $fa_i$ uses the domain transportation theory (see Section 3) to calculate an optimal resource allocation proposal $rap_{ik}$ based on all available resources that under $fa_i$'s management (Line 14). After the calculation of $rap_{ik}$, $fa_i$ submits $rap_{ik}$ to $da$ if current time $tca$ has not exceeded task proposal deadline. The task proposal deadline is calculated by function $PDLINE(tas_k\.dea, eve.sev)$, which can be defined by local emergency departments based on the detail of $tas_k$ and $eve$. After the submission of $rap_{ik}$, $da$ adds $rap_{ik}$ to $RAP_k$.

**Step 4:** (Lines 20-23) After $da$ receives $tas_k$'s resource allocation proposals from all facility agents in $FA_k$ or $tas_k$'s proposal deadline has been reached, $da$ checks whether $RAP_k$ has enough resources for $da$ to generate a final resource allocation plan to complete $tas_k$. If the resources are enough, the process goes to Step 6. Otherwise, the process goes to Step 5.

**Step 5:** (Lines 24-25) If $RAP_k$ does not have enough resources to complete $tas_k$, $da$ expands its original communication area $da\.com$ by double to contact more facility agents, and then the process goes back to Step 2.

**Step 6:** (Lines 27-29) If $RAP_k$ has enough resources to complete $tas_k$, $da$ uses domain transportation theory (see Section 3) to generate an optimal resource allocation proposal $rap_k^\ast$ for $tas_k$ based on all resources in $RAP_k$, which is represented by $RED_k$ (Line 24). Finally, $da$ informs relevant facility agents to execute $rap_k^\ast$ and remove $tas_k$ from eve.TAS. If there are more tasks in eve.TAS, the process repeats Step 4, otherwise the process ends.

5 Experiment

In this section, experimental results are presented and the performance of the proposed resource allocation approach is analysed. The experiments focus primarily on testing the resource allocation time, money expenditure and cost of an event when employing the proposed optimal resource mapping algorithm.

The benchmark of the experiments is the decentralised ambulances allocation approach proposed by Beatriz et al. [5], which uses an auction mechanism based on trust and time to assign ambulances to emergency events.

5.1 Experimental Setting

In the experiments, the proposed resource allocation was tested in two different scenarios, which are: (1) resource allocation for an event with the proposed optimal resource allocation approach and (2) resource allocation for an event with Beatriz's ambulances allocation approach.

Although Beatriz's approach was specifically designed to allocate ambulances, but it is suitable to be used to allocate different kinds of mobile resources. Therefore, only mobile resources were required by the two scenarios' experiments. Furthermore, both two scenarios were conducted over an $1000 \times 1000$ grid. The experiments in each scenarios were repeated for
1000 times and the average resource allocation cost, time and money expenditure of an event was recorded. The resource allocation cost was calculated by Equation 1. For each time of the experiment, the resources’ parameters were randomly regenerated based on Table 1, but the parameters of the event (Table 2) and its tasks (Table 3) remain same.

Table 1  Parameters for Resource’s Setting

<table>
<thead>
<tr>
<th>rty</th>
<th>vlo</th>
<th>ava</th>
<th>vel</th>
<th>exp</th>
<th>fun</th>
</tr>
</thead>
<tbody>
<tr>
<td>mobile</td>
<td>[0, 1000]</td>
<td>{0, 1}</td>
<td>[20 − 200]</td>
<td>[1 − 100]</td>
<td>{ ambulance, fire truck, police car }</td>
</tr>
</tbody>
</table>

Table 2  Parameters for Event’s Setting

<table>
<thead>
<tr>
<th>con</th>
<th>SER</th>
<th>elo</th>
<th>sev</th>
<th>TAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>fire</td>
<td>{ fire &amp; rescue, medical, police }</td>
<td>[500, 500]</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 3  Parameters for Task’s Setting

<table>
<thead>
<tr>
<th>No.</th>
<th>ser</th>
<th>dea</th>
<th></th>
<th>TR</th>
<th>tr.rty</th>
<th>tr.fun</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>fire &amp; rescue</td>
<td>30 min</td>
<td>15</td>
<td>mobile</td>
<td>fire truck</td>
<td></td>
</tr>
<tr>
<td>Task 2</td>
<td>medical</td>
<td>40 min</td>
<td>10</td>
<td>mobile</td>
<td>ambulance</td>
<td></td>
</tr>
<tr>
<td>Task 3</td>
<td>police</td>
<td>50 min</td>
<td>10</td>
<td>mobile</td>
<td>police car</td>
<td></td>
</tr>
</tbody>
</table>

5.2 Experimental Result and Analysis

Fig. 2 Experimental Results

The experimental results are shown in Fig. 2. As we can see from Fig. 2 (a), Beatriz’s approach (Scenario 2) requires slight less resource allocation time for an event comparing with our approach (Scenario 1). Nevertheless, from Fig. 2 (b) and Fig. 2 (c), we can see that our approach outperforms Beatriz’s approach significantly in terms of money expenditure (1086 versus 1329) and allocation cost (9850 versus 18100). This is mainly because in Beatriz’s approach, resource allocation time is used as the primary criterion to determine the optimal allocation of resources, but our approach takes both resource allocation time and resource money expenditure into consideration. Another potential reason behind the results is that Beatriz’s resource allocation approach was based on auction mechanism and the relationships between contract agents are competitive. However, for most emergency situations, it is more reasonable for agents to act cooperatively rather than competitively. In our approach, a deployment agent uses the domain transportation theory
to generate the optimal solution based on the resource allocation proposals from multiple facility agents, so the optimal solution is a combined solution by integrating the advantages of each facility agent’s proposal. By doing so, our approach can effectively minimise the required resource money expenditure and allocation cost by not compromising too much in terms of resource allocation time.

6 Conclusion and Future Work

In this paper, an agent-based decentralised resource allocation approach was proposed to handle an emergency event in metropolitan regions. In order to efficiently search an optimal resource allocation proposal for an event that requires different resources, the proposed approach first creates a set of tasks based on the emergency services required by the event. Then, domain transport theory is used to search the optimal resource allocation proposal for each task. The proposed approach was designed to handle multiple resource allocation tasks simultaneously and it can be used for emergency events in different domains. The experimental results indicate the good performance of the proposed approach in terms of resource allocation cost. Our future work will focus on handling concurrent emergency events and incorporating a resource coordination mechanism for resource contention problems between events.

References

A proto-type of a portable ad hoc simple water gauge and real world evaluation

Takanobu Otsuka¹, Yoshitaka Torii, and Takayuki Ito

Nagoya Institute of Technology, Japan,
otsuka.takanobu@nitech.ac.jp,
http://www.itolab.nitech.ac.jp

Abstract. In recent years, with landslides caused by flooding of rivers due to heavy rain, the flooding and destruction of houses has increased. With regard to the structure of the river and height above sea level, how to completely prevent flood damage by internal water is difficult. To that end we have created a prototype of a portable ad hoc simple water gauge for the purpose of installation in the field in collaboration with Nippon Koei Co., Ltd. In particular, in portable wireless sensor networks, communication reliability is greatly influenced by the surrounding environmental factors and by communication node movement. Furthermore, features and issues of the portable ad hoc simple water gauge system require that it is operated at the actual disaster site, and human observation is also required for production. In this paper, the task of building an ad hoc simple water gauge system that assumes movement was carried out with implementation of portable wireless sensor networks.

Keywords: Wireless Sensor Network, Field Informatics, Disaster Informatics

1 Introduction

In recent years, with landslides caused by flooding of rivers due to heavy rain, the flooding and destruction of houses has increased. With regard to the structure of the river and height above sea level, how to completely prevent flood damage by internal water is difficult. At present, it is only possible to address the operation of drainage devices typified by car-to-pump submerged location. However, to drain floodwater as quickly as possible, it is necessary to understand the full scope of floodwater with water-level data of the flooding points throughout. To that end, we have built a prototype of a portable ad hoc simple water gauge for the purpose of installation in the field in collaboration with Nippon Koei Co., Ltd. In particular, in portable wireless sensor networks, communication reliability is greatly influenced by the surrounding environmental factors and by communication node movement. Therefore, it differs from the fixed wireless sensor networks. The radio wave condition is affected buildings and vegetation, as well as changes in the environment and movement of the nodes. Furthermore, features and issues of the portable ad hoc simple water gauge system require that it is operated at the actual disaster site, and human observation is also required for production. In this paper, the task of building an ad hoc simple water gauge system that assumes movement was carried out with implementation of portable wireless sensor networks.
The rest of the paper is organized as follows. Section 2 introduces previous studies and the relative position of our research. Section 3 describes the configuration of our system. Then, we describe the suggested design guidelines for overcoming problems in section 4, and in section 5 we show experimental results of the prototype system. Finally, section 6 gives the report summary and touches on future challenges.

2 Related Work

2.1 Research Trends in WSN

In recent years, research has been conducted widely on wireless sensor networks (WSN), the Internet of Things (IoT), and Machine to Machine (M2M) core technology [1]. Nodes of the WSN acquire sensor data such as temperature, illumination, and acceleration, and can constitute a "multi-hop ad hoc network" to transfer the obtained data to a bucket relay system using radio waves [2] [3]. WSN, which can configure an autonomous network simply by the placement of nodes, can reduce the installation work required in field use. Also, it is possible to capture real-world dynamics, tracking and monitoring of the natural environment, and this has been widely studied as a promising application of WSN.

Wireless sensor networks are a number of slave units and a sink node for aggregating information from the sensing nodes, and also have multi-hop relay capable router nodes. They are beginning to be widely used in fields such as smart home technology. One famous research application is MOTE [4], which was developed at UC Berkeley in 1998. In recent years, it has been standardized as IEEE802.15.4, and actively studied with a Zigbee network. However, use in Japan has been severely limited compared to other countries. Technical issues such as construction authentication, unnecessary communication modules and limited communication distance must be overcome.

In recent years, individual interoperable environment information acquisition devices have been sold widely for research applications and home use. One wireless sensor system is MEMSIC's Eko system [5]. Research has also been conducted on agricultural uses [6], but showed poor scalability. Moreover, as a consumer product, there is Davis’s Vantage series [7], but it is only capable of operating in a dedicated application, Thus, it is not suitable for large-scale data collection. As an example of environmental information gathering by experimental WSN, it is possible to install a large number of observation nodes in an active volcano [8] for the purpose of predicting eruptions.

We have developed a sensor network device and server application that can collect high-density environmental data on a large scale. In addition, research [9] into the optimization of sensor arrangement for the purpose of reducing power consumption has been conducted. Also, the construction of sensor node applications applying agent technology has been investigated [10]. Furthermore, with the aim of manageability of the operational aspect, it is possible to prepare nodes of the program according to the purpose. Research exists on facilitating the construction of a sensor network according to applications [11]. Network topology depending on circumstances has been studied as well [12]. Thus, the study of WSN as a framework for performing flexible measurement is active.
2.2 Studies on radio wave quality stabilization of WSN

The most sought-after characteristic of WSN is network stability. Studies have been widely conducted for the purpose of continuous data collection and optimization of the placement of fixed sensor networks. For example, using a radio wave level (RSSI values) of the topology formed by the nodes, studies to optimize node arrangement in a predefined space have been conducted [13]. To capture the node as a cluster, research [14] to optimize the communication route has been carried out. Other research focused on optimizing the position of the relay station for multi-hop between nodes, and reducing communication errors [15]. Another study examined arranging a plurality of sink nodes to optimize their positions [16].

However, much research focuses on optimization for a "closed environment". In securing the communication quality of the mobile-type WSN, we must consider how the ambient environment changes dynamically. Thus, this research is difficult to apply directly to the portable sensor networks that operate in the field. In particular, the sensor network is constructed in a wide range for the purpose of obtaining environmental information. The need to communicate over a distance in agriculture and in river management operations means the radio wave state around the installation site must be measured in advance. Preliminary surveys and techniques for installation in consideration of the placement of the measurement nodes are common. However, installation of the above for portable sensor networks that can operate in the field is not realistic. In recent years, and landslides caused by flooding of the river due to heavy rain, the number of such flooding of houses has increased. With regard to internal water damage, the structure of the river, above sea level and, such as where a landfill, how completely prevent flood damage by internal water is difficult. At present, it is only possible addressing of operating the drainage devices typified by car to pump submerged location. However, in order to quickly drained work as possible, it is necessary to understand the full flooding water by the water level data of the flooding point throughout. To that end we have done a prototype of a portable ad hoc simple water gauge for the purpose of installation in collaboration with Nippon Koei Co., Ltd. In particular, in a portable of the wireless sensor networks, communication reliability is greatly influenced by the surrounding environmental factors by communication node movement. Therefore, differ from the fixed wireless sensor networks. It, the radio wave condition and buildings around which changes due to the movement, is the effect of such plants. Furthermore, features and issues required portable ad hoc simple water gauge system is operated in the actual disaster site, it is necessary to determine standing worker eyes for performing production.

In this paper, it is possible to organize the task of ad hoc simple water gauge system that assumes the movement, was carried out implementation of portable wireless sensor networks.

3 Implementation of portable ad hoc simple water gauge

3.1 Requirement for portable ad-hoc water gauge capability

In recent years, and landslides caused by flooding of the river due to heavy rain, the number of such flooding of houses has increased. With regard to internal water dam-
age, the structure and of the river, above sea level and, such as where a landfill, how completely prevent flood damage by internal water is difficult. At present, it is only possible addressing the drainage devices typified by car to pump submerged location. However, in order to quickly drained work as possible, it is necessary to understand the full flooding water by the water level data of the flooding point throughout. A mobile ad-hoc level gauge is mounted on the sink node on the pump truck to perform drainage of flooded areas, and to aggregate the water level information from the measurement nodes installed elsewhere. From that information, the water level conditions in the flooded area can be visualized, which will help reduce the time required to drain floodwaters. In addition, the cooperation with the pump future vehicle integrated management system is intended to improve the efficiency of wastewater work using a pump wheel with a small number of operators. We show a system outline in Fig. 1. For

![Fig. 1. System Overview](image)

- Sensor data stream -
  - Water depth sensor
  - Voltage, current sensor
  - GPS module
  - Link quality
  - Number of hops

the system to be operational in reality with high reliability, sensor node research applications are required. Further, because it differs with the agricultural WSN and outdoor fixed operations, the following three problems are present.

**Challenge of communication reliability** Since the sink node and the measurement nodes are installed anywhere, it is difficult to consider the pre-positioning. Therefore, not only the communication distance is required to check the redundancy of the ad hoc network.

**Challenges of systems** Since the measurement node itself moves, the current position in real time must be known, and it is necessary to clearly display the water level change.
Challenges in operation  Because it is operated at disaster sites, ease of installation in the field and maintainability of the time of failure are essential. Also, unlike the stationary WSN, external factors affecting the signal quality change dynamically. Thus, the radio wave condition must be measured in advance, and optimization of placement is difficult. Therefore, it is required that all communication nodes act as relay stations in the ad hoc network. Thus, rather than the hardware side, and identifying the measurement nodes and the relay node by setting the connection whether the water level gauge on the server software. Moreover, given the task of mounting the measurement and router nodes located at a disaster site, it is necessary to reduce the effort of the operator. Therefore, ease of installation and visualization of communication node location for acquiring water level information are high priorities.

4 Proposal and implementation of technical problem solving

In this section, we address the problems listed in Section 3, and detail implementation of the solutions. First, we describe the communication reliability challenges. The communication reliability involves the problem of how to overcome communication failure due to the influence of buildings and forest zones in a disaster site. When compared to indoor sensor networks, outdoor sensor networks must be installed in areas where buildings are crowded, and radio wave interference is likely. In particular, in our development of portable wireless sensor networks, communication reliability is greatly influenced by the surrounding environmental factors and by communication node movement. Field tests must be conducted to solve the signal quality for the portable sensor networks. We also propose a recommendation algorithm for installation position. Also, in the WSN research field in Japan using ad-hoc networks, Digi’s Xbee series is often used. In studies we conducted, significant advantages in power consumption and communication distance were not seen. We adopt Tokyo Cosmos’s TWE-Strong [18], which has software that can allow all nodes to operate as both relay and measurement nodes. Nodes can be switched to relay or sensor status from a web UI with respect to the conversion of the depth data to the water level, which is displayed as the water level data by using the offset amount from the reference value. Therefore, even for places where radio wave conditions are bad, it is possible to ensure the reliability of communication by relaying data by the relay node. Also, for the problem of the system plane, with respect to locating the node, it is possible to send the current position to a GPS module. The sensor management screen is shown in Fig. 2. In addition, we want to take advantage of the knowledge [5] [17] of the systems operational in agriculture. In the event of sensor failure, the modular structure of the circuit makes module replacement easy. The circuit board shown in Fig. 3 was implemented. Communication nodes are mounted on an easily removable facilitating structure that is fixed to a tripod or similar device that is installed on the ground using a clamping tube. Also, solar cells that are required in the case of long-term operation are mounted with a hinge structure, thereby achieving efficiency of operation as required. We show the communication node appearance in Fig. 4. As described above, by performing the actual design and implementation from an operational point of view, a system easy to use for operators was implemented in hardware and software.
Fig. 2. Switching of measurement and relay node by web UI

Fig. 3. Developed circuit board
5 Field operation experimental results

In this section, we show the results of operational experiments performed at Mishima, Shizuoka on February 12, 2015. The experiment fields are the branch line and main line to the confluence of the Kano River. It has experienced three episodes of flooding over the past 10 years, and is designated as a flood hotspot on the hazard map provided by the government. This experiment was carried out under the supervision of the Ministry of Land, Infrastructure and Transport, Chubu Maintenance Division. The equipment and its layout in the experimental field are shown in Fig. 5. In experiments, assuming that inland water damage occurs, we observed the actual level due to installation and immersion of depth sensor-type portable ad hoc water gauges using the actual river. In this experiment, the following performance criteria were confirmed.

- After local arrival, immediately confirmed that it is possible to operate.
- Confirmation of redundancy due to the ad hoc network when a part of the communication path becomes disconnected.
- Easiness of tablet viewing by web UI.

In addition to these items, experiments were carried out in an actual river watershed. The portable ad hoc water gauge is mounted with the following procedure.

1. Fixed the communication node by tube clamp on a tripod.
2. Writes tow anchor rivers, to irrigation canals.
3. Turned on by the mounted pilot wire water depth sensor to the anchor.
4. Connected the water depth sensor to communication node.

With the above procedure, it is possible to measure water level anywhere. Installation landscape and procedures are shown in Fig. 6. We established a gateway that is connected to the 3G communication network that assumes the pump car mounted on the
As shown above, the result of the operation experiment of this system, commissioned by the Ministry of Land, Infrastructure and Transport, was successful. A portable ad hoc water gauge was developed for assessing internal water damage at a disaster site. In addition, ease of use and maintenance in the field were confirmed. Also, radio propagation in urban areas is a challenge, and the impact of the type waves reflected by the outer walls of buildings is a factor. There are many parameters affecting communication availability. Currently, we are conducting an ongoing operational test of the system. Nodes developed in this study have been used to continuously monitor water levels in Mishima, Shizuoka, Misono district. We have been able to continuously verify the reliability of long-term behavior of the hardware and software of our system.

1. After unloading equipment, water level can be measured in 5 minutes.
2. Even when we intentionally cut off a portion of the communication path for 30 seconds (one sampling period), it communicatively detours to other routes.
3. The web UI shows real data without having to explain it to many of the participants.

As shown above, the result of the operation experiment of this system, commissioned by the Ministry of Land, Infrastructure and Transport, was successful. A portable ad hoc water gauge was developed for assessing internal water damage at a disaster site. In addition, ease of use and maintenance in the field were confirmed. Also, radio propagation in urban areas is a challenge, and the impact of the type waves reflected by the outer walls of buildings is a factor. There are many parameters affecting communication availability. Currently, we are conducting an ongoing operational test of the system. Nodes developed in this study have been used to continuously monitor water levels in Mishima, Shizuoka, Misono district. We have been able to continuously verify the reliability of long-term behavior of the hardware and software of our system.

![Fig. 5. Overview of test field](image-url)
Fig. 6. Installation landscape and procedures
6 Conclusion

In this paper, we implemented a portable ad hoc water level meter that can be operated in the event of a disaster. We also developed a portable sensor network system that is highly robust against obstacles and changes in the environment. In particular, in portable wireless sensor networks, communication reliability is greatly influenced by the surrounding environmental factors and by communication node movement. Furthermore, it is possible to operate the system at an actual disaster site. As a result, installation can be completed in 5 minutes, it was possible to build a system capable of rapid water level measurement at the time of a disaster. We also checked for redundancy of the ad hoc network, and it was possible to demonstrate the utility of this system. In the future, we will address the challenges of long-term investment, and planning the size of the communication nodes. Next year, the system will be operated and tested with actual pump trucks in the field, hopefully yielding an even more useful system. Also, as wireless communication standards continue to advance, we will upgrade our system accordingly. Thus, upgrades of the communication module, increased power efficiency, etc. will be achieved through continuing research and development.

acknowledgment

Part of this study was conducted in cooperation with the Ministry of Land, Infrastructure and Transport, Chubu Regional Development Division as a part of the "Nobi Plain drainage plan". In addition, this study is a collaboration with Nippon Koei Co., Ltd.
References

Chapter 1
Exploiting Vagueness for Multi-Agent Consensus

Michael Crosscombe and Jonathan Lawry

Abstract  A framework for consensus modelling is introduced using Kleene’s three valued logic as a means to express vagueness in agents’ beliefs. Explicitly borderline cases are inherent to propositions involving vague concepts where sentences of a propositional language may be absolutely true, absolutely false or borderline. By exploiting these intermediate truth values, we can allow agents to adopt a more vague interpretation of underlying concepts in order to weaken their beliefs and reduce the levels of inconsistency, so as to achieve consensus. We consider a consensus combination operation which results in agents adopting the borderline truth value as a shared viewpoint if they are in direct conflict. Simulation experiments are presented which show that by applying this operator to agents chosen at random (subject to a consistency threshold) from a population, with initially diverse opinions, results in convergence to a smaller set of more precise shared beliefs. Furthermore, if the choice of agents for combination is dependent on the payoff of their beliefs, this acting as a proxy for performance or usefulness, then the system converges to beliefs which, on average, have higher payoff.

Key words:  Agent-Based Modelling · Many-Valued Logics · Belief Aggregation · Consensus

1.1 Introduction

Reaching a consensus by agreeing a shared viewpoint or position is a fundamental part of many multi-agent decision making and negotiation scenarios. In this paper we argue that by exploiting vagueness in the form of explicitly borderline cases we can define an operator for belief combination which not only allows a population of agents to reach consensus but also results in them adopting, on average, a more useful set of beliefs. The basic intuition underlying this operator is that conflicting agents can agree to allocate borderline truth values to propositions about which they hold inconsistent beliefs. For example, two individuals, one of which believes that ‘Cameron is an effective prime minister’ whilst the other believes that ‘Cameron is ineffective’, may agree, in some circumstances, to adopt the shared view that ‘Cameron is borderline effective/ineffective’.
Of course, beliefs about the world do not exist in isolation but inform and influence our decisions and actions. From this perspective, some sets of beliefs are more positive or useful than others, resulting in better long term performance, perhaps by making the individuals concerned richer, happier or just better able to survive. More generally, in a multi-agent context, different beliefs result in different actions, collecting different payoffs. In this paper we present simulation studies which show that implementing our proposed operator across a population of agents, initially holding diverse beliefs, results in convergence to a smaller subset of more precise shared opinions. Furthermore, under the assumption that better performing agents, i.e. those with higher payoff, are more likely to interact to reach consensus, we show that the range of beliefs obtained at steady state are on average better, i.e have higher payoff, than the agents’ initial beliefs. The formalism adopted here is that of Kleene’s three valued logic and the operator investigated has been proposed for single propositions in [2] and extended to multi-propositional languages in [5].

An outline of the paper is as follows: Section 1.2 gives a brief overview of consensus modelling. Section 1.3 introduces Kleene logic and the three valued consensus combination operator. Section 1.4 describes simulation experiments in which agents are selected at random to form a consensus provided that they are sufficiently consistent with one another. In section 1.5 we introduce a payoff function for beliefs, so that the payoff of a particular set of beliefs acts as a proxy for the performance of an agent holding those beliefs. We then adapt the experiments described in section 1.4 so that the probability of an agent being selected for consensus is proportional to their payoff. Finally, in section 1.6 we give some discussions and conclusions.

1.2 Background and Related Work

A number of models for consensus have been proposed in the literature which have influenced the development of the framework described in this paper. [3] introduced a model for reaching a consensus involving a weighted, global updating of beliefs, iterating until an agreement is reached. In DeGroot’s model, agents assign a weight distribution to the population before forming a new opinion. By applying their assigned weights to the beliefs of others, an agent can control the influence that others have on their own beliefs.

As an alternative to DeGroot’s model, the Bounded Confidence (BC) model introduced in [4] provides agents with a confidence measure. An agent quantifies their level of confidence in their own opinions and are then able to limit their interactions to those agents who possess similar beliefs if they are highly confident (small bounds), or extend the range of possible interactions if the agents possess low confidence (large bounds). In this model agents do not a priori assign weights to the beliefs of others, but instead determine such weightings based on similarity and on their own confidence levels. This is similar in essence to the inconsistency threshold that we introduce in section 1.3, but applied on an individual basis.
The Relative Agreement (RA) model [7] then extends the Bounded Confidence model to allow agents to assign weights to the beliefs of others by quantifying the extent of the overlap of their respective confidence bounds. By having agents declare a confidence interval for their beliefs, the model then restricts interactions to those pairs of agents with overlapping intervals. Consequently, agents are only required to assess their own beliefs and are not required to make explicit judgements about those of other agents. [7] also moved to a model of pair-wise interactions to better capture social interactions of individuals, the latter being a setting in which group-wide updates to beliefs are unintuitive in that they do not reflect typical social behaviour.

A fundamental difference between our approach and the above models is that we use Kleene’s three valued logic to represent beliefs in a propositional logic setting, rather than identify opinions with real values or intervals. [2] have shown that through use of a three-state model for networked consensus of complete graphs, nodes converge to a consensus much faster and with greater accuracy when compared to a restrictive binary model. In the sequel we extend this approach to a more general setting involving larger languages and incorporating a measure of payoff for beliefs.

### 1.3 A Three Valued Consensus Model

In this section we introduce Kleene’s three valued logic [10] as a model of explicitly borderline cases resulting from the inherent vagueness of propositions. We adopt a propositional logic setting as follows: Let $\mathcal{L}$ be a finite language of propositional logic with connectives $\land$, $\lor$ and $\neg$, and propositional variables $\mathcal{P} = \{p_1, \ldots, p_n\}$. Also, let $S\mathcal{L}$ denote the sentences of $\mathcal{L}$ generated by recursively applying the connectives to the propositional variables in the usual manner. A Kleene valuation then allocates truth values $0$ (false), $\frac{1}{2}$ (borderline) and $1$ (true) to the sentences of $S\mathcal{L}$ as follows:

**Definition 1.1. Kleene Valuations**

A Kleene valuation $v$ on $\mathcal{L}$ is a function $v: S\mathcal{L} \to \{0, \frac{1}{2}, 1\}$ such that $\forall \theta, \varphi \in S\mathcal{L}$ the following hold:

- $v(\neg \theta) = 1 - v(\theta)$
- $v(\theta \land \varphi) = \min(v(\theta), v(\varphi))$
- $v(\theta \lor \varphi) = \max(v(\theta), v(\varphi))$

The truth table for Kleene valuations are shown in table 1.1.

It is sometimes convenient to represent a Kleene valuation $v$ by its associated orthopair $(P, N)$ [5], where $P = \{p_i \in \mathcal{P} : v(p_i) = 1\}$ and $N = \{p_i \in \mathcal{P} : v(p_i) = 0\}$. Notice that $P \cap N = \emptyset$ and that $(P \cup N)^c$ corresponds to the set of borderline propositional variables.

Kleene valuations have been proposed as a suitable formalism in which to capture explicitly borderline cases as resulting from inherent flexibility in the definition of
vague concepts in natural language [11, 12]. For example, consider the proposition ‘Ethel is short’. For the concept short, we might identify a lower height threshold \( h \) below which any height is classed as being absolutely short, and similarly there may be an upper threshold \( h \) above which any height is absolutely not short. If Ethel’s height lay between \( h \) and \( h \) then this would result in a borderline truth value for the statement ‘Ethel is short’.

It is important to note that the middle truth value \( \frac{1}{2} \) is not intended to represent epistemic uncertainty, but rather explicitly borderline cases resulting from the inherent vagueness of natural language propositions. Hence, if we say that the statement ‘Ethel is short’ is borderline true/false we are not saying that the truth or falsity of this proposition is unknown. Instead we are indicating that Ethel’s height is a borderline case of the predicate short. In order to emphasise the difference between the epistemic and the borderline interpretation of three valued logic it is helpful to think in terms of conditioning. For instance, if we learn that it is unknown whether or not Ethel is short, then this provides us with no new information about her height. In contrast, learning that Ethel is borderline short does provide us with new information about Ethel’s height, namely that it lies on the borderline between short and not short. A more comprehensive discussion of these issues can be found in [1]. A consequence of using this interpretation of the middle truth value is that in the current paper we only model consensus for sets of propositions which admit borderline cases. In other words, our approach can be used for propositions such as ‘Ethel is short’ but not, for example, for the proposition ‘Ethel is strictly less that 1.4 metres tall’.

The following three valued consensus operator was described in detail in [5]:

**Definition 1.2. Consensus Operator**

Let \( v_1 \) and \( v_2 \) be Kleene valuations on \( L \) with associated orthopairs \((P_1,N_1)\) and \((P_2,N_2)\). Then the consensus \( v_1 \odot v_2 \) is the Kleene valuation with the orthopair \(((P_1\cap N_2) \cup (P_2\cap N_1), (N_1\cap P_2) \cup (N_2\cap P_1))\)

The corresponding truth table for this operator is shown in table 1.2. From this we can see that the operator preserves the non-borderline truth values 0 or 1 except in the case of a direct conflict i.e. when one agent has truth value 1 and the other 0. In this case both agents adopt the middle truth value \( \frac{1}{2} \). Alternatively, from definition 1.2 we can think of \( \odot \) as an operator which initially weakens both opinions so as to remove direct inconsistencies, before then combining them.
1.4 Simulation Experiments based on Random Selection of Agents

We introduce simulation experiments in order to investigate the convergence properties of the three valued logic operator when implemented across a multi-agent system. The experimental set up is loosely based on those proposed in [7] and [9], although our representation of opinions is quite different with beliefs taking the form of Kleene valuations on \( \mathcal{L} \), rather than vectors of bounded real numbers.

We will consider two distinct initialisations of the beliefs of a population of agents. The random three valued initialisation allocates the truth values 0, \( \frac{1}{2} \), and 1 to each agent and each propositional variable at random i.e. with probability \( \frac{1}{3} \) for each truth value. In contrast, the random Boolean initialisation only allocates the binary truth values 0 and 1, each with a probability of \( \frac{1}{2} \). This latter initialisation

\[
\begin{array}{c|ccc}
\circ & 1 & \frac{1}{2} & 0 \\
1 & 1 & 1 & \frac{1}{2} \\
\frac{1}{2} & 1 & \frac{1}{2} & 0 \\
0 & \frac{1}{2} & 0 & 0 \\
\end{array}
\]

Table 1.2: Truth table for the consensus operator.

We now introduce two measures that will be used throughout the subsequent simulation experiments.

**Definition 1.3. A Measure of Vagueness**

Let \( v \) be a Kleene valuation on \( \mathcal{L} \) with orthopair \((P, N)\) and \( n \) propositional variables. Then we measure the vagueness of \( v \) by the proportion of propositional variables which it classifies as being borderline. That is:

\[
V(v) = \frac{|(P \cup N)^c|}{n}
\]

**Definition 1.4. Inconsistency Measure**

Let \( v_1 \) and \( v_2 \) be Kleene valuations on \( \mathcal{L} \) with corresponding orthopairs \((P_1, N_1)\) and \((P_2, N_2)\). Then we define the inconsistency measure of \( v_1 \) and \( v_2 \) to be the proportion of propositional variables which are in direct conflict between the two valuations i.e. \( v_1(p_i) \neq \frac{1}{2}, v_2(p_i) \neq \frac{1}{2} \) and \( v_1(p_i) = 1 - v_2(p_i) \). That is:

\[
I(v_1, v_2) = \frac{|(P_1 \cap N_2)| + |(P_2 \cap N_1)|}{n}
\]

In the sequel we will propose a threshold \( \gamma \in [0, 1] \) on inconsistency so that valuations \( v_1 \) and \( v_2 \) can be combined only if \( I(v_1, v_2) \leq \gamma \).
will be required in section 1.5 in order to directly compare the proposed three valued combination operator with a similar two valued operator. In this section we will use the random three valued initialisation in order to investigate the extent to which the three valued operator results in convergence to a shared set of opinions across the population of agents.

We set a fixed maximum number of 50,000 iterations\(^1\). At each time step a pair of agents are selected at random from the population. An inconsistency threshold value \(\gamma \in [0, 1]\) is set, so that for any pair of agents with respective valuations \(v_1\) and \(v_2\), if \(I(v_1, v_2) \leq \gamma\) then both agents replace their beliefs with the consensus valuation \(v_1 \odot v_2\), while if \(I(v_1, v_2) > \gamma\) then no combination is performed and both agents retain their original beliefs. For \(\gamma = 1\) we obtain what is equivalent to the totally connected graph model described in [2], in which any pair of agents can combine their beliefs, whilst taking \(\gamma = 0\) corresponds to the most conservative scenario in

\(^1\) In preliminary experiments we found that 50,000 was an upper bound on the number of iterations required for the system to reach steady state across a range of parameter settings.
which only absolutely consistent beliefs can be combined. The parameters for the
simulation experiments are then as follows:

- Population size: 100
- Language size i.e. $|\mathcal{P}| = n$: 5, 10, 50, 100
- Initial beliefs: Random three valued.
- Inconsistency threshold: $\gamma \in [0, 1]$. 

Figures 1.1 and 1.2 show the results for the experiments after 50,000 iterations. In each case the plots show mean values with error bars representing standard deviation across 100 independent runs of the simulation. Figure 1.1 shows the average vagueness determined by taking the mean value of $V(v)$ (definition 1.3) across the population. Note that for a random three valued initialisation of beliefs we expect a mean vagueness value of $\frac{1}{3}$ at the start of the simulation. As the threshold $\gamma$ increases then the average vagueness decreases to zero, so that for $\gamma \geq 0.3$ we are left with almost entirely crisp (i.e. Boolean) opinions. In general the more conservative the combination rules (i.e. requiring higher levels of consistency) then the more it is that vague beliefs are maintained in the population. Figure 1.2 shows the number of distinct valuations (i.e. different opinions) remaining in the population after 50,000 iterations. Again this decreases with $\gamma$ and for $\gamma > 0.4$ agents have on average converged to a single shared belief. This is consistent with the analytical results presented in [2] for the single propositional, $\gamma = 1$ case.

1.5 Simulation Experiments Incorporating a Payoff Model

In this section we extend the simulation framework described in section 1.4 to allow for different payoffs for different beliefs. As outlined in section 1.1, payoff is introduced as a proxy for performance, and is motivated by the intuition that different beliefs result in different actions which then, over time, lead to different levels of performance. Here we adopt an abstract simplification of this process in which each Kleene valuation is allocated a real valued payoff. Then, instead of being selected at random for combination, an agent is picked from the population according to a probability which is proportionate to the payoff value of their beliefs. The idea, then, is that agents with better or more useful opinions will be more successful and furthermore, it will be these successful agents who will be most likely to need to reach a consensus between them.

Here the underlying intuition is that, in real systems it is the most successful agents, with the highest payoff values, who are most likely to find themselves in conflict with one another, and who will most benefit from reaching an agreement. We adopt a simple summative payoff model in which each propositional variable $p_i$ is allocated a value in the range $[-1, 1]$, denoted $f(p_i)$, and the payoff for a valuation with orthopair $(P, N)$ is then calculated as follows:
Another perspective on this type of payoff function is as follows: For each propositional variable $p_i$, a truth value of 1 results in a payoff $f(p_i)$ (which can be either positive or negative), a truth value of 0 results in the opposite signed payoff $-f(p_i)$, and a borderline truth value $\frac{1}{2}$ results in a neutral payoff of 0. The payoff value for a Kleene valuation $v$ is then simply taken to be the sum of the payoffs for each propositional variable under the truth values allocated by $v$.

Based on payoff values we define a probability distribution over the agents in the population according to which the probability that an agent with beliefs $(P, N)$ is selected for possible consensus combination is proportional to $f(P, N) + n$. At each iteration a pair of agents are selected at random according to this distribution. For each such pair the inconsistency measure (definition 1.4) is evaluated and either both
the valuations are replaced with the consensus valuation, or both are left unchanged, depending on the threshold $\gamma$ as in section 1.4. The parameters for the simulation experiments are as follows:

- Population size: 100
- Language size: 5
- Initial beliefs: Random Boolean.
- Inconsistency threshold: $\gamma \in [0, 1]$

Notice that here we are initialising the beliefs as random Boolean valuations (see section 1.4). This allows us to make a direct comparison between the performance of the three valued combination operator and a similar two valued operator. For the latter we assume that only binary truth values are available to represent an agent’s beliefs. In this context, in order for two agents with conflicting truth values for $p_i$ (i.e. one 0 and the other 1) to reach consensus, we propose that they simply agree to pick one of the truth values at random e.g. by tossing a fair coin. Table 1.3 gives the truth table for the operator in which directly conflicting truth values leads to a stochastic outcome.

The focus on simulations with 5 propositional variables is intended to increase the number of opinions relative to the size of the population, in order to achieve a good distribution of valuations. For example, a language size of 5 allows for 32 possible Boolean valuations. With a population of 100 agents, it is therefore very likely that each opinion will occur at least once. In comparison, a language size of 10 produces $2^{52}$ possible Boolean valuations which severely decreases the probability of an opinion being present in a population of the same size.

Figures 1.3, 1.4 and 1.5 show the results for simulation experiments with agent selection based on payoff. The results shown are mean values with error bars taken over 100 independent runs of the simulation. Figure 1.3 shows the average population payoff after 50,000 iterations given as a percentage of the maximal possible payoff value i.e. the payoff for the valuation $(P_N)$ where $P = \{ p_i : f(p_i) > 0 \}$ and $N = \{ p_i : f(p_i) < 0 \}$. For both the binary and the three valued operators we show results for simulations in which agents are selected according to payoff (three-valued, Boolean) and at random as in section 1.4 (three-valued random, Boolean random). We see that for all values of $\gamma$, the three valued operator with payoff based selection outperforms all of the other approaches. For the former we can also see that average payoff increases with $\gamma$. In contrast, for the other approaches, including the payoff operator with payoff based selection, the mean of the average population payoff

---

2 As a result of this Boolean initialisation, a language size of 5 now produces a total of $2^5$ (32) possible valuations, as opposed to $3^5$ (243) possible valuations.
remains close to 0 after 50,000 iterations. Figure 1.4 shows the mean number of distinct valuations across the population of agents after 50,000 iterations. All four versions of the operators converge on a small set of shared beliefs for sufficiently large $\gamma$. For $\gamma \geq 0.4$ the mean number of distinct valuations is less than 5 while for $\gamma \geq 0.8$ it is 1. Figure 1.5 shows a trajectory of how the number of distinct valuations varies with each iteration when $\gamma = 0.7$. We can see that both the three-valued models converge quickly (in just over 2000) iterations while the Boolean models require considerably longer to converge (over 20,000 iterations).

1.6 Conclusions

In this paper we have explored the use of Kleene’s three valued logic as a framework in which to model multi-agent consensus formation. We have proposed a three valued combination operator, the intuition behind which is that conflicting binary truth values are replaced with a borderline (middle) truth value. A number of simulation experiments have been presented employing this operator. These can be divided into two main categories. For the first type of experiments, agents are selected at random from the population and form a consensus valuation providing that the level of inconsistency of their respective opinions is below a threshold parameter $\gamma$. Otherwise they do not form a consensus and instead retain their current opinions. For these experiments we found that there is convergence to a smaller subset of shared opinions across the population. For higher $\gamma$ values there is convergence on average to a single shared opinion and furthermore this opinion is crisp i.e. it admits no borderlines. For intermediate values of $\gamma$ the system convergences to a small set of opinions which to some extent remain vague.

In the second type of experiments a payoff function over beliefs is introduced, and agents are selected for possible combination with probability proportional to the payoff value of their current beliefs. Here we compare the three value operator
with a similar stochastic Boolean operator. We find that the three valued operator with payoff based agent selection results in convergence to a smaller shared set of beliefs with significantly higher average payoff than that of the initial population. The Boolean operator does not perform well in this context and does not result in a significant increase in average payoff, which instead remains close to 0 after 50,000 iterations.

The results of the payoff based experiments show how a three valued model for consensus provides a number of improvements over a traditional Boolean model. Firstly, we have shown that the introduction of Kleene valuations to capture the inherent vagueness of propositions does not, in the long run, lead to the mass adoption of borderline truth values as a result of conflict occurring in the population. Instead, we have seen how vagueness is reduced at lower $\gamma$ values, and at higher $\gamma$ values the population converges towards completely crisp opinions on average, admitting no borderline cases. In addition to this, we can see that the introduction of a payoff based model drives consensus towards those valuations which result in higher payoff on average. By selecting pairs of agents based on their perceived success, we can achieve an increase to overall payoff in a small number of iterations, compared to no significant increase in payoff for the Boolean model. Therefore, we have shown that the three valued approach incorporating a payoff model can drive convergence across the population towards more successful opinions.

We suggest that the experiments presented in this paper show the potential of using three valued logic in consensus modelling. There is also significant scope to extend the research presented in several new directions. For example, the above studies concern consensus defined at the level of propositional variables. However, in many cases agents will be most concerned to reach agreement about a relevant set of compound statements. For example, they may need to reach agreement about a particular set of conditional statements, or equivalences. Hence, an important question is that of how best to extend our proposed consensus model so as to be applicable to compound logical expressions. Another significant question concerns uncertainty. Suppose that in addition to vagueness agents also quantify their uncertainty about beliefs. [5] propose an extension of the three valued framework in which agents’ beliefs are represented by a probability distribution over Kleene valuations. Ongoing research concerns the design of simulation studies in which to evaluate the convergence and payoff based performance of this extended model. Finally, it would be interesting to consider extensions to the operator which allows for consensus between groups rather than just pairs of agents.

**Acknowledgements**

This research is partially funded by an EPSRC PhD studentship as part of a doctoral training partnership (DTP).
References

Chapter 1
Selecting Robust Strategies Based on Abstracted Game Models

Oscar Veliz and Christopher Kiekintveld

Abstract  Game theory is a tool for modeling multi-agent decision problems and has been used to great success in modeling and simulating problems such as poker, security, and trading agents. However, many real games are extremely large and complex with multiple agent interactions. One approach for solving these games is to use abstraction techniques to shrink the game to a form that can be solved by removing detail and translating a solution back to the original. However, abstraction introduces error into the model. We study ways to analyze games that are robust to errors in the model of the game, including abstracted games. We empirically evaluate several solution methods to examine how robust they are for abstracted games.

Key words: Game Theory, Agent Based Simulation, Agent Based Modeling, Abstraction

1.1 Introduction

Game theory is widely used for analyzing multi-agent decision problems including auctions [12], security [17, 1], Poker [14], and many others. A game-theoretic analysis begins by specifying a formal model of the decision problem, including the actions players can choose, the information available to the players, and the utilities they receive for different outcomes. Once the model is specified it can be analyzed using Nash Equilibrium or any of the other numerous solution concepts proposed in the literature [16].

There has been extensive research on solution concepts, but less attention has been given to the problem of specifying game models. The model is typically assumed to be given as a starting point, and common knowledge to all players. This is
problematic for several reasons. Games that model real world interactions are often complex, with huge numbers of possible strategies and information states. Formal specifications of these games may be intractable to solve, even for modern supercomputers (or humans). A second problem is that the game itself may be ambiguous and difficult to formalize. Players may not have the same understanding of the possible actions, the payoffs, and the knowledge and beliefs of the other players about the game.

We argue that most game models should be considered abstractions that approximate more complex situations. An abstracted game model captures some—but not all—of the relevant detail about the strategic decision. In some cases, abstraction may also be used intentionally as part of the process of analyzing a complex game. For example, work on artificial agents for Texas Hold ’em poker has made extensive use of a methodology that uses abstraction to shrink the size of the game tree before applying an (approximate) equilibrium solver to compute a strategy for the game [14].

We are interested in better understanding the effect of abstraction in game-theoretic analysis. In particular, we focus on the strategy selection problem: how should an agent choose a strategy to play in a game, based on an abstracted model of the game? This problem has three interacting components: (1) the method for abstracting the game, (2) the method for selecting a strategy based on the abstraction, and (3) the method for mapping this strategy back to the original game. This approach has been studied fairly extensively for poker, which is a 2-player, zero-sum game. However, much less is known about how abstraction interacts with strategy selection in more general games.

The main contributions of our work are as follows. First, we specify a model of the strategy selection problem when players use asymmetric abstractions as a meta-game. In this model players can use different methods for abstracting the game, solving the game, and reverse-mapping the solution. We introduce a collection of specific methods for abstracting and solving games; these are intended to be representative of the most popular methods used in the literature. Finally, we present the results of extensive simulation that evaluate the candidate abstraction and solution methods on different classes of games. Our results lead to several unique observations as well as identifying solution methods that are more robust than others to error introduced by abstraction.

1.2 Abstraction Meta-Games

We first introduce a formal model that can be used to study the situation where players select strategies based on abstracted game models. Our model is based on the meta-game framework introduced by Kiekintveld et al. [11], which focused on situations where players received noisy observations of the same underlying game and had to select strategies based on these observations. The situation where players use abstractions is similar in that the players make strategy choices based on imperfect abstractions of the game. Opposing players may also use different abstractions
which may cause problems for solution concepts that rely on coordination (such as Nash equilibrium).

An example of an abstraction meta-game is shown in Figure 1.1. In this example, we have two players who are playing the one-shot normal form game shown at the top of the figure; this is the base game. Each player has four possible actions in the original game, and the payoffs are listed in the matrix. Each player uses a different (unspecified) abstraction method to reduce the size of the game to only two actions for each player, as shown. Now the players analyze these smaller games to select a strategy to play. Here, both of the small games can be solved using dominance to find a unique Nash equilibrium. The players select these strategies (A and H) to play. However, when these strategies are played in the base game they result in the outcome $-10, -10$, which is the worst possible payoff for both players!

In addition to illustrating the structure of the abstraction meta-game, this example shows one of the possible problems with using abstraction in game analysis. Note that we could replace the payoffs $-10, -10$ with arbitrarily small values. By playing according to a (dominant strategy) Nash equilibrium of the smaller abstracted game, the payoffs the players actually receive are less than the payoffs they expect to receive by an unbounded amount. They can also be lower than the equilibrium payoffs in the original game, or even the maxmin payoffs by an arbitrary amount.

This emphasizes the problem of selecting strategies based on abstracted games. Many existing applications of automated abstraction (e.g., in Poker [8, 14, 15, 10]) still analyze the abstracted games to find an (approximate) Nash equilibrium, and play the original game using the equilibrium strategy for the abstracted game. Much of this work focuses on zero-sum games where Nash equilibrium is equivalent to the pessimistic minmax solution concept, and has a unique value. However, for general-sum games it is clear from our example that the simple approach of analyzing abstracted games using solution concepts that assume perfect knowledge of the game model is highly problematic. Even in the case of poker, there have been pathologies shown in using abstraction [18], which has led to the exploration of alternative methods for playing games using abstraction that are not based on playing according to an approximate Nash equilibrium [7, 9].
1.3 Abstraction Methods

We define an abstraction method as a function that maps one normal-form game into a second (smaller) normal-form game. A good abstraction should be simpler than the original, but also retain as much information about the strategic interaction as possible. We identify two broad categories of abstractions that are common in the literature: strategy elimination and strategy combination.

Strategy elimination abstractions remove some strategies completely to reduce the size of the game. The well-known method of iterated elimination of dominated strategies [16] is one example of this approach. Weakening the criteria for dominance so that weakly or approximately dominated strategies are also removed leads to even smaller games at the cost of potentially removing some equilibria of the original game [3].

Abstractions methods based on strategy combination simplify games by merging multiple strategies into a single representative strategy. In the extreme case where two strategies have exactly identical payoffs for a player in every outcome merging these strategies results in a lossless abstraction. The idea can be extended to merge strategies that are only similar rather than identical, but the result is a lossy abstraction. A common approach in the literature is to use clustering algorithms to merge similar strategies into clusters. In extensive form games, clustering can also be applied to information states [8]. A different type of similarity between strategies can be used to solve a game by reducing it to sub games [4].

Our goal in this paper is not to develop novel abstraction methods, or to exhaustively evaluate the many existing techniques. We are interested in the interaction between abstractions and the solution methods applied to abstracted games. Therefore, we selected two simple abstraction methods for our initial study that are representative of the broad categories described above. We also do not try to guarantee any bounds on these general abstractions as shown in Figure 1.1 with potentially unbounded error. It is also the case that the simulated agents may not be using an equilibrium or convergent strategy.

1.3.1 TopN Abstraction

The first abstraction method we consider is TopN. This method creates a smaller game by selecting a subset of size N of the strategies for each player to form the abstracted game. For each strategy in the game, we first calculate the expected payoff of the strategy against a uniform random opponent strategy. We then select the N strategies with the highest expected payoffs, breaking ties randomly. The abstracted game is simply the game where players are restricted to playing only the N selected strategies; the payoffs are unchanged. Since each strategy in the abstracted games is also a strategy in the original game the reverse mapping of the strategies back to the original game is trivial.
\subsection{1.3.2 K-Means Clustering Abstraction}

The second abstraction method we use is \textit{KMeans}, which is representative of strategy combination methods. This method uses the \textit{k}-means clustering algorithm to group strategies into clusters based on similarity of their payoffs. Each strategy is represented in the clustering algorithm by a vector of the payoffs for the strategy in every outcome that can result when the strategy is played. We cluster the strategies for each player separately, following the standard \textit{k}-means algorithm. Each strategy is initially assigned randomly to one of the \textit{k} clusters. The centroid is calculated for each cluster, and then the Euclidian distance between every strategy and every cluster centroid is calculated. Strategies are re-assigned to the closest cluster, ensuring that no cluster becomes empty. This process iterates until there are no further changes in the clusters. We run the clustering 100 times with different initial clusterings, and select the one with the smallest maximum distance between any strategy in a cluster and the centroid.

Once the strategies are clustered we create the abstracted game as follows. Each cluster maps to a pure strategy in the abstracted game (so the number of strategies is controlled by the \textit{k} parameter used in \textit{k}-means). The payoffs for each outcome in the abstracted game are computed by averaging the payoffs for all of the outcomes in the cluster. In other words, we assume that players will play each strategy in a given cluster with equal probability. The reverse mapping also assumes that players play strategies in the same with uniform random probability. For example, if a strategy in the abstracted game places a probability of 0.5 on playing a given pure strategy, this probability would be distributed uniformly over all of the strategies that comprise that cluster in the strategy used to play the original game.

\subsection{1.4 Candidate Solution Methods}

The task of our solution methods is to select a strategy to play in a game. We consider several candidate solution methods for selecting strategies in abstracted games. All of these are based on known solution concepts or simple heuristics for playing games, and they are intended to provide a diverse pool of plausible strategies to evaluate. Of particular interest are several solution concepts that originate in behavioral game theory and have been shown to predict human behavior better than Nash equilibrium and related concepts. We hypothesize that humans may be adopting these types of strategies in part because they are more robust to ambiguity and uncertainty.

\textbf{Uniform Random (UR)}: Play each pure strategy with equal probability.

\textbf{Best Response to Uniform Random (BRU)}: Play the pure-strategy best-response to UR. It is equivalent to a level-1 strategy in the cognitive hierarchies model [2].

\textbf{Nash Equilibrium (MSNE)}: We use Gambit [5] logit solver to calculate a sample Nash equilibrium in mixed strategies and play according to this strategy.

\textbf{Epsilon-Nash Equilibrium (ENE)}: For every pure-strategy profile, we first calculate the maximum that value ($\varepsilon$) that any player can gain by deviating to a different
pure strategy. We select the profile with the smallest value of $\varepsilon$ (the best approximate Nash equilibrium in pure strategies) and play the associated pure strategy.

**MaxMin:** Play the strategy that maximizes the worst-case payoff for the player.

**Fair:** This heuristic strategy focuses on outcomes that are “fair” in that there is a small difference between the payoffs for the players. For every strategy profile we calculate the difference between the payoffs and select an outcome that minimizes this difference. Ties are broken in favor of outcomes with a higher sum of payoffs, and then randomly.

**Social:** This strategy plays according to the outcome that maximizes the sums of the payoffs for all players. If there are ties the strategy plays a uniform random strategy over the strategies in the tied outcomes.

**Quantal Response Equilibria (QRE):** Quantal Response Equilibrium [13] originated in behavioral game theory. It incorporates a model of noisy best-response where players use a softmax decision rule instead of strict best-response. A logistic function is normally used to specify this response, and it has a parameter $\lambda$ that interpolates between a uniform random strategy when $\lambda = 0$ and a best response as $\lambda \to \infty$. A QRE is defined similarly to a Nash equilibrium, except that both players play noisy best-responses to each other for some value of $\lambda$. QRE has been shown to provide a better fit for experimental data on human behavior in some games [13]. QRE has also been shown to have more robust strategy choices that NE in situations where players make choices based on noisy observations of an underlying game [11]. We compute a QRE using Gambit [5] and play a best-response to the predicted QRE strategy of the opponent.

**Cognitive Hierarchies (CH):** Cognitive Hierarchies [2] also originates in behavioral game theory. It models a recursive style of reasoning where level-0 agents play a uniform random strategy, level-1 agents play a best response to the level-0 agents, level-2 agents play a best response to a mixture over level 0 and 1 agents, etc. Agents at each level use a Poisson distribution, based on a parameter $\tau$, to predict the probabilities of playing agents at lower levels, and play a best response to this mixture.

**Quantal Level-k (QLK):** This method combines the features of QRE and CH. It uses a recursive reasoning model identical to CH, except that in place of the best-response at each level agents play a noisy best-response using the same logit function used in QRE. It has parameters for both $\lambda$ and $\tau$. We play a best-response to the predicted strategy of the opponent, based on the specified level of reasoning.

### 1.5 Experimental Methodology

We run simulations using the meta-game framework from Section 1.2. Our testbed allows us to run tournaments with different combinations of (1) classes of base games, (2) abstraction methods, and (3) solution methods. The simulation first generates a random game from the class as the base game. Each meta-strategy is a combination of an abstraction method and a solution algorithm. We calculate the
strategies selected by each meta-strategy and then play a round-robin tournament among these strategies. The payoffs are based on the strategy profile that is played in the original game.

The result of the tournament is a payoff matrix for the meta-game where each meta-strategy can be played by either player. To estimate this matrix for a class of base games we average the payoffs over a large number of sampled games. We can analyze this payoff matrix to generate several performance metrics for the meta-strategies. Average payoffs are quite biased, in that they can reward strategies that only do well against other weak strategies. Instead, we present our results using stability and average regret measures. Stability is a measure of how close the pure strategy profile where all players use the same meta-strategy is to being a pure Nash equilibrium. It is calculated by finding the maximum gain for deviating to another meta-strategy from this profile. Any value less than or equal to zero indicates that the profile is a pure equilibrium of the estimated meta-game. Average regret is the average of the regret for playing a given meta-strategy against each other meta-strategy in the estimated game.

We use three classes of randomly-generated games: general sum, zero sum, and logical games. General sum and zero sum are generated using GAMUT [6]. Players have 20 pure strategies, and payoffs are real numbers in the range [-100,100]. Logical games have more structure and should be more amenable to abstraction. Pure strategies are based on choosing either 0 or 1 for a set of boolean variables. Payoffs are based on randomly generated oppositions with varying numbers of clauses, literals, and values. If a proposition is true for a given outcome in the game, the payoff associated with the proposition is added to the payoff for that player. The payoffs are additive over all true propositions for a given outcome. Finally, payoffs are normalized to the range [0,100]. We also could not possibly evaluate every level of abstraction so we picked several interesting abstraction levels. An abstraction level of 10 means that there are now 10 actions in the game, Top10 or 10 clusters, and 20 indicates these are the original 20 pure strategies with no abstraction.

1.6 Results

Our experiments are designed to test the relationships between abstraction methods and solution concepts across different classes of games, and with varying levels of abstraction. In total, we experiment with three different classes of games, two types of abstraction, four levels of abstraction, and more than 30 different solution methods (including multiple parameter settings for QRE, CH, and QLK). The main results span 24 different tournaments with 500 sample games played in each one.

The full data set is difficult to visualize, so we will present results focusing on selected subsets of agents to illustrate key points. We focus on measures of stability and regret here as we believe they are the most informative, but we have also looked at measures of average payoffs and exploitability. Average payoffs can be misleading for agents that exploit weak players but perform poorly against strong players.
Exploitability does not differentiate well among the meta-strategies, since almost all of them are highly exploitable in the worst case.

We first present a set of results that demonstrates how we compare the different parameter settings for QRE, CH, and QLK. The purpose of this analysis is to discover which parameter settings are strongest. In later results we will include only best parameter settings for each solution method to aid in visualization.

Figures 1.2 and 1.3 show results for QRE with different values of the $\lambda$ parameter. Each plot shows results for all three classes of games. There are separate plots for each type of abstraction. The x-axis shows the level of abstraction where the values are the number of actions in the abstracted game. The point on the far left corresponds to no abstraction, and more information is lost for data points further to the right. The y-axis represents the stability ($\varepsilon$) value for each solution method.

Lower stability values are better. In particular, any value less than or equal to 0 indicates that a solution method is a pure-strategy Nash equilibrium, and if all players were using this method none of them would benefit by deviating to any other solution method. These stabilities are calculated with respect to the full data set, meaning that players are allowed to deviate to any solution method, not just the ones in the figure.

In general sum and logical games, the best QRE parameter settings are closer to playing a best-response to a more random opponent than playing a best-response to an opponent closer to a Nash equilibrium. For example, the very low parameter setting of $\lambda = 0.05$ performs very well, and this setting is the closest to uniform random. Zero-sum games appear to behave quite differently where playing closer to an equilibrium strategy performs better.
We perform a similar set of experiment to determine the parameterizations of CH and QLK agents for the values of $k$, $\tau$, and $\lambda$. We omit them due to space considerations, but use a similar procedure to identify the most effective parameter settings to visualize in the later plots. The values of $\tau$ considered are 1.5, 2.5, 5, 7.5, and 9, the values of $k$ are 0, 1, 2, 3, 5, 10, and 20, and the values of $\lambda$ are 0.8, 1.5, 2.0, 5.0, and 20. For QLK agents we also consider both version that play the raw QLK strategy and ones that play a best response to the strategy of the opponent for a particular level. One results to note from this analysis is that for CH agents with the same $\tau$ but different levels of reasoning there is not much difference between agents with mid-level $k$ versus high level $k$. The main variations between these agents occur at lower levels. However, higher levels of $k$ typically have better, more robust performance.

We now turn to analysis of the complete set of agents. The following results are from a fully symmetric tournament of 500 games for each class of games and 30 agents which include the best parameter settings for QRE, CH, QLK, and the other main solution concepts. We selected the best parameter settings for each of the QRE, CH, and QLK variants based on the results of the previous analysis to visualize in these results along with the other agents.

Fig. 1.4 Key for Figures 1.5 and 1.6
when using the KMeans and TopN abstractions. We begin by noting some general trends in the results. Although we don’t show the results for logical games all of the stability and regret values are very low compared to both general-sum and zero-sum games. There is also very little variation among each of the strategies. This indicates that these games are much easier to play well in, even when they are abstracted. In most cases, agents are more stable for cases without any abstraction, and less stable but converging to the same range as abstraction increases. Zero-sum games behave somewhat differently; the overall stability is worse, and the stability of many of the agents actually improves (surprisingly) as abstraction increases.

There is no solution concept that is clearly the best across all of the settings tested. Instead, the best strategy depends heavily on both the class of games and the type of abstraction being used. The UR and Fair agents do poorly in almost all cases. The ENE and Social agents perform surprisingly well, particularly when using the TopN abstraction and in the general-sum and logical games. These agents focus on coordinating on high payoffs, so it is interesting that even these simple strategies are able to do so well even in cases with abstraction error. For the cases with KMeans abstraction and for zero-sum games, approaches close to equilibrium solvers perform better particularly MSNE and the QRE and QLK.

While this pattern of results fails to identify a clear winner among the common solution methods for how to select strategies in abstracted games, the results are still quite interesting. They clearly show that both the game class and abstraction method interact strongly with the applied solution concept, and simply finding a NE is not the best approach. There is also an opportunity to develop significantly improved solution concepts that account for these factors unlike current practices.
The final experiment we present directly compares the performance of the two different types of abstractions. In this experiment, players can use either abstraction, which leads to even more asymmetry in the abstractions players are using. In Figure 1.7 we show the average payoffs for the different solution methods in a tournament that combines agents using both abstractions. We use only the best parameter settings for the QRE, CH, and QLK agents. This tournament was run on general-sum games, using the highest level of abstraction (from a game size of 20 down to a game size of 5). For each solution method we show the average payoffs achieved when using the two abstraction methods side by side. Interestingly, the TopN abstraction method outperforms KMeans in combination with every one of the solution methods, often quite substantially. The TopN abstraction plays a role in coordinating agents on high-payoff outcomes independent the solution algorithm.

1.7 Conclusion

Our results demonstrate that using abstraction to solve games is a complex endeavor, and the type of abstraction, the solution methods used to analyze the abstracted games, and the class of games all have a strong influence on the results. Many of the strongest results using abstraction to analyze large games (e.g., Poker) have focused on zero-sum games. One of the most interesting observations from our results is that abstraction often works very differently in zero-sum games than it does general-sum games or the more structured logical games. In particular, solution methods based on finding Nash equilibrium seem to work much better in zero-sum games than they do in the other classes of games in our experiments. Another important observation from our experiments is that Nash equilibrium often does not perform well in cases where abstraction is used as part of the solution process. It is still effective when the games are zero-sum, but in the other cases it was not robust to the introduction of error based on game abstraction.

We also found that the specific method used for generating abstractions has a strong impact on the results. One very interesting result was that TopN was sometimes able to increase the payoffs for the agents in comparison to the case without any abstraction. However, TopN is a symmetric abstraction when both players use it, while KMeans is asymmetric. Some methods like the social agent performed much better when using the symmetric TopN abstraction than when using the asymmet-
This kind of interaction is very important to understand in greater depth if we are to make effective use of abstraction as part of game-theoretic analysis. Our model of abstraction meta-games provides a formal model for studying this type of interaction, and our simulations have resulted in several interesting observations that provoke many additional questions about the use of abstraction in solving games.

References

Simulating and Modeling Dual Market Segmentation Using PSA Framework

Quan Bai, Jiamou Liu and Ziheng Wei

School of Computer and Mathematical Sciences
Auckland University of Technology, New Zealand
{quan.bai, jiamou.liu, bys7090}@aut.ac.nz

Abstract. Market segmentation aims to divide the market into subsets taking into account multiple factors such as consumer needs, interests, and tastes. The outcome of market segmentation is heavily influenced by the interactions between consumers and commodities. Given these interactions, we employ the Propose-Select-Adjust (PSA) framework. Our approach is distributed and can be applied to large and dynamic market data set. The experimental results suggest that the proposed approach is a promising technique for supporting intelligent market segmentation.

Keywords: Market segmentation, PSA framework, simulation

1 Introduction

Market segmentation amounts to one of the most important marketing strategies. It aims to divide a broad market into smaller subsets, taking into account different factors such as customers’ interests, tastes, and priorities, revealing important information regarding lifestyles, geographic differences, and other demographic and economical phenomena. With this information, a business defines its target customers, enabling more accurate orientation of marketing plans, allowing more efficient use of resources and hence increasing profits.

Market segmentation is a complex and data intensive task. In order to pinpoint the exact correlation between the above factors and the markets, an analyst needs to take into account a huge amount of market history and transaction records. Hence there has been a significant interest in computational approaches that support market segmentation. Numerous recent works applied data mining techniques to this problem [11, 10, 12, 1]. One common and central goal of these works is to build an intelligent tool which, through analysing large amounts of market data, computes an appropriate segmentation of the market.

Research on market segmentation exhibits a diverse landscape. For example, Miguís, et. al. recently use lifestyle information derived from market segmentation to analyse customer purchasing power [11]. In another work, the same authors also investigate the design of product promotion strategies based on market segmentation results [10]. Market segmentation has also been applied in customer loyalty management[12].

Market segmentation can be made from two perspectives: consumers and commodities. On one hand, the process categorises consumers into different subgroups. On the other hand, it also distinguishes clusters of commodities. These
two types of segmentation strongly influence each other. For example, a specific group of consumers may prefer a range of products which results in the forming of a new commodity segment; and the introduction of a specific range of products may also enforce the emergence of a consumer segment. The relation between consumers and commodities are interactive, highly dynamic and often subtle. It is therefore an important question to analyse interactions between consumers and commodities.

Based on the above observations, we propose the dual market segmentation problem. Our goal is to design a computational framework for segmenting – at the same time – both the consumer and commodity markets, taking into account their mutual influence.

In our previous work [9], we proposed a decentralised computational framework, Propose - Select - Adjust (PSA), for solving network problems. In this framework, each node of the graph acts as an individual computational unit which performs three procedures in cyclic order: Propose, Select and Adjust. The nodes are acting in an asynchronous manner while each node only has access to its local information. Through a series of case studies and experiments, we demonstrated the approach is a promising solution for simulating community formation, and hence detecting communities in a large dynamic network [8].

In this paper, we extend the PSA framework on the bipartite market network connecting consumers and commodities. We suggest that PSA can be applied to market segmentation. This approach has several advantages:

- Firstly, PSA is a technology for smart simulation of group behaviors in a network. Thus it simulates the formation of segments within the consumers and commodities. This simulation naturally captures the mutual influence between consumers and commodities, which is otherwise hard to define.
- Secondly, PSA as a distributed framework, can scale to handle large amount market data, while computation is kept local.
- Thirdly, as a PSA cell runs continuously, the PSA cells monitors the changing market, and detects emergence of market segments.

We support our claims above with experimental results on both synthesised and real data.

The rest of the paper is organised as follows. Section 2 formulates the dual market segmentation problem using a bipartite graph model. Section 3 describes the PSA framework and extend it to the bipartite dual network of consumers and commodities. Our new framework provides two perspectives: The local perspective segments consumers and commodities based on their links. The global perspective, on the other hand, captures the interactions between the segments identified in the local perspective. Our experimental results are presented and discussed in Section 4. Some related works are reviewed in Section 5. Finally the paper is concluded in Section 6.

2 Problem Formulation

A bipartite graph is a pair \((V_0 \cup V_1, E)\) where \(V_0\) and \(V_1\) are two sets of nodes with \(V_0 \cap V_1 = \emptyset\), and the edge relation \(E\) is a subset of the Cartesian product.
Abstractly we use a bipartite graph to model the problem domain. In particular, we may view $V_0$ as a set of consumers, and $V_1$ as a set of commodities in a market; an edge $(v, p) \in V_0 \times V_1$ denote that consumer $v$ purchases the commodity $p$. Hence this bipartite graph captures a collection of purchasing records of all consumers in $V_0$. We point out that the terms “consumers” and “commodities” are used in a broad sense: these are two abstract concepts which captures any interacting objects in a market-like context.

This bipartite graph model involves a dual network: On one hand, from the perspective of a consumer, its outgoing edges indicate its purchased commodities – intuitively, two consumers have similar purchasing patterns, if they purchase similar commodities. On the other hand, from the perspective of a commodity, its incoming edges indicate its consumer base – intuitively, two commodities have similar consumer base, if they are purchased by similar groups of customers.

The main problem of the paper is to compute a segmentation of this dual network. For consumers, a segmentation amounts to dividing the market into groups of consumers with similar purchasing patterns which are affected by consumers’ tastes, life styles, etc. For commodities, a segmentation amounts to aggregation of correlated products on the market. Formally, we model market segmentation as follows:

**Definition 1 (Dual segmentation).** Let $G = (V_0 \cup V_1, E)$ be a bipartite graph. A dual segmentation is a pair $(\sim_0, \sim_1)$ where $\sim_i$ is an equivalence relation on $V_i$ for $i \in \{0, 1\}$. An $\sim_i$-equivalence class is called a $V_i$-segment.

We observe that, in the context of a market, the two components of a dual segmentation are interrelated. For example, consumers with a similar lifestyle tend to purchase a similar group of products, which in turn causes correlation among these products. Therefore, our goal is to provide a uniform approach for computing a dual segmentation, in which the segments of consumers influence segments of commodities, and vice versa.

This approach builds the two networks involved in the market: the consumer network, and the commodity network. A similarity function on a set $S$ is a function $\omega : S \times S \to \mathbb{R}$. The key ingredients of our approach are two similarity functions, $\omega_0, \omega_1$, that are defined on the two sets $V_0$ and $V_1$, respectively. Our similarity functions $\omega_0, \omega_1$ are based on Jaccard distance. Given a bipartite graph $G = (V_0 \cup V_1, E)$, for $u \in V_0$, $v \in V_1$, define

$$E(u) = \{v' \in V_1 \mid (u, v') \in E\} \text{ and } E^{-1}(v) = \{u' \in V_0 \mid (u', v) \in E\}.$$  

The similarity $\omega_0(v_1, v_2)$ between $v_1, v_2 \in V_0$ is the Jaccard distance

$$\omega_0(v_1, v_2) = \frac{|E(v_1) \cap E(v_2)|}{|E(v_1) \cup E(v_2)|} \quad (1)$$

The similarity $\omega_1(v_1, v_2)$ between $v_1, v_2 \in V_1$ is the Jaccard distance

$$\omega_1(v_1, v_2) = \frac{|E^{-1}(v_1) \cap E^{-1}(v_2)|}{|E^{-1}(v_1) \cup E^{-1}(v_2)|} \quad (2)$$

Using these two similarity functions, two networks can be constructed:
Definition 2 \((V_i\text{-networks})\). Let \(G = (V_0 \cup V_1, E)\) be a bipartite graph, and \(\omega_0, \omega_1\) be two similarity functions on \(V_0\) and \(V_1\) respectively. For any \(i \in \{0, 1\}\), the \(V_i\text{-network} is an undirected graph \((V_i, E_i)\) where
\[
E_i = \{(u_1, u_2) \in V_i^2 \mid \omega_i(u_1, u_2) \geq \alpha_i\}
\]
\(\alpha_i\) is the local similarity threshold.

We call \(V_0\text{-network} and \(V_0\text{-segments} as consumer network and consumer segments, and call \(V_1\text{-network} and \(V_1\text{-segments} as commodity network and commodity segments.

The simulation of segmentation involves two steps: local and global. In the local step, we construct for each \(i \in \{0, 1\}\), a set of disjoint \(V_i\text{-segments} \(C_i = \{C_{i,0}, ..., C_{i,\ell}\}\) where each \(C_{j,i} \subseteq V_i\) and \(\bigcup C_j \subseteq C_i = V_i\). Intuitively each \(C_j,0\) represents a group of consumers who share some common interests. Thus each \(V_i\text{-segment} ideally should be a maximal clique in the \(V_i\text{-network}\). Since finding maximal cliques is a well-known NP-hard problem, we use the following notions to approximate a clique-like subgraph.

Definition 3 \((\text{Local core})\). In a graph \(G = (V, E), N(v)\), the closed neighbourhood of \(v\), is a set \(\{u \mid (u, v) \in E\} \cup \{v\}\). A \(k\)-core in a graph is an induced subgraph where all nodes have degree at least \(k\). The core number of a vertex \(v\) is the largest \(\kappa(v)\) such that \(N(v)\) contains a \(\kappa(v)\)-core. For \(v \in V\), the local core of \(v\) is the set
\[
K(v) = \{u \in N(v) \mid |N(u) \cap N(v)| \geq \kappa(v) - 1\}
\]

The result of the local step consists of a set \(C_0\) of consumer segments and a set \(C_1\) of commodity segments. From the perspective of consumers, several consumer segments can share common interests towards the same commodity segments. From the perspective of commodities, several commodity segments are deployed to the same consumer segments. This calls for possible combinations of these already computed segments. Therefore, in the global step, we take the edge set \(E(C_0, C_1, E^H) = \{(u, v) \in C_0 \times C_1 \mid (u, v) \in E\}\), and construct a bipartite graph \(G^H = (C_0 \cup C_1, E^H)\) where
\[
E^H = \left\{(C_1, C_2) \in C_i \times C_{1-i} \mid \frac{|E(C_1, C_2)|}{|C_i|} \geq \beta_i, i \in \{0, 1\}\right\}
\]
and \(\beta_i\) is the segmentation selection rate, which determines the importance of the \(V_2\)-segment \(C_2\) to the \(V_1\)-segment \(C_1\).

Similarly to (1) and (2), we define global similarity functions \(\omega_0^H\) and \(\omega_1^H\) on \(C_0\) and \(C_1\), respectively. The graph \(G^H\) has corresponding dual networks: \(C_0\text{- and} C_1\text{-network where}
\[
E_i^H = \{(C_1, C_2) \in C_i^2 \mid \omega_i^H(C_1, C_2) \geq \gamma_i\}
\]
where \(\gamma_i\) is the global similarity threshold. The similarity \(\omega_i^H(C_1, C_2)\) between \(C_1, C_2 \in C_i\) where \(i \in \{0, 1\}\) is the Jaccard distance.
3 The PSA Framework For Segmentation Simulation

3.1 General Description

We extend the Propose-Select-Adjust (PSA) framework to simulate market segmentation based on a bipartite graph model. Intuitively, each node in the graph is a *cell* which carries out certain tasks independently. The PSA framework describes how to implement a single *cell*. The framework is inspired by the decision making process among a group of people: Imagine a group of individuals trying to decide on a partitioning of the group, where every member would belong to one and only one subgroup. The constraint is that each individual only sees local information about her own connections; no knowledge is shared among all members. Thus individuals can only make self-centred judgements and decide on the people that she would like to be with. Under this constraint, the following procedures can ensure the group arriving at a collective decision:

- **Propose**: Each person individually decides a list of people whom she would like to include in her own community. She then sends an invitation to everyone on the list to form a community. Here we implicitly assume that a person makes an invitation to herself.
- **Select**: After all invitations are received, the person evaluates the quality of each proposed community, selects and accepts the best invitation.
- **Adjust**: Once a person accepts an invitation, she then updates her own community according to the accepted proposal. After every individual finishes this step, the whole group would have been divided into a number of communities, and thus a clustering is formed.

More precisely, the crucial ingredients in the definition of a PSA cell are a set of proposals and a preference relation defined as follows:

**Definition 4 (Proposal and Preference).** A proposal of $v \in V$ is a set $P_v \subseteq N(v)$. All possible proposals in a system $G = (V,E)$ are related by a preference relation $\preceq$ which is required to be a linear ordering.

Hence any precise specification of the PSA system involves defining a specific proposal for each cell, and a preference relation. In our bipartite graph market model, each consumer and each commodity will act as a PSA cell, which carries out computation based on its local information: for a consumer this information contains all commodities that consumer chooses, for a commodity, this information contains all consumer that chooses this commodity. Each cell performs three phases of computation repeatedly, which are roughly described below:

In the **Propose** phase, every cell $v$ prepares and announces a proposal $P_v$ which is received by all nodes contained in the proposal. In particular, as the proposal $P_v$ will automatically contain the node $v$ itself, we also regard $v$ as having received the proposal. Then in the **Select** phase, every cell $v$ takes the collection of proposals it receives, and chooses a most preferred proposal according to the relation $\preceq$. Suppose $v$ selects the proposal $P_u$ proposed by $u$. In the **Adjust** phase, $v$ observes the action of $u$ and makes the two possible actions:
1. If \( u \) chooses its own proposal \( P_u \) (in this case, its own proposal is seen as the most preferred option), then \( v \) joins the group of \( u \).

2. If \( u \) chooses the proposal \( P_w \) of some node \( w \neq u \), then \( v \) does not join the group of \( u \), and disregard the proposal from \( u \).

After finishing the Adjust phase, every cell would go back to Propose phase and iterate the above three phases again. This enables dynamic computing: as more market data arrives, every cell would make corresponding, spontaneous change to its own computation, but recomputing a proposal and adjusting its solution. Moreover, assuming the market data is stable (so no change occurs anymore), then every PSA cell in this simulation will eventually reach a stable solution.

The simulation must handle both the local and global steps of segmentation. We need one PSA system to compute the local segmentation of consumers and commodities, and another PSA system to compute the global segmentation. Hence, we use two types of PSA cells: local PSA cells and global PSA cells.

### 3.2 Local PSA Cells

In a system \( G = (V_0 \cup V_1, E) \), we compute the corresponding \( V_0 \)- and \( V_1 \)-networks \((V_0, E_0) \) and \((V_1, E_1) \) as described above. In an undirected graph, a local core represents a set of nodes that are similar to each other. Therefore each cell \( v \in V \) makes a proposal \( P_v = K(v) \). In the graph \( G = (V, E) \), for any set \( C \subseteq V \), we use \( N(C) \) to denote the union \( \bigcup_{u \in C} N(u) \). We utilise the following density measures:

- The intra cluster density of \( C \) is the percentage of the number of edges in \( C \) over all possible internal edges; in other words:
  \[
  d_{\text{intra}}(C) = \frac{2 \times |E \cap C|}{|C| \times (|C| - 1)}
  \]  

- The inter cluster density is the percentage of the number of edges connecting \( C \) with an outside node over all possible links from \( C \) to outside nodes; in other words:
  \[
  d_{\text{inter}}(C) = \frac{|(E \cap N(C)) \setminus (E \cap C)|}{|C| \times |N(C) \setminus C|}
  \]  

Combining these two factors, we obtain a utility of a set \( C \):

\[
 d(C) = d_{\text{intra}}(C) \times (\kappa(v) + 1) - d_{\text{inter}}(C) \times |C|
\]

Note that intuitively, a set \( C \) has higher utility if it is dense, has relatively large cardinality, and sparsely connected to nodes outside of it. To define the preference between proposals, we set \( C_1 \preceq C_2 \) whenever \( d(C_1) \geq d(C_2) \).

As described above, every cell \( v \) makes a proposal \( P_v \) to its neighbours in the Propose step. Then every cell analyses all received proposals and chooses one most preferred proposal according to the preference relation \( \mu \) defined above.

### 3.3 Global PSA Cells

After constructing the local segmentation, we obtain a new bipartite graph \( G^H = (C_0 \cup C_1, E^H) \) whose nodes are segments computed by the local PSA cells. We
use another set of PSA cells to compute a global segmentation, where each cell stands for a local segment in \( C_0 \cup C_1 \). Similarly to the local perspective, the global perspective also consists of a dual networks: \((C_0, E^H_0)\) and \((C_1, E^H_1)\). For each \( C \in C_i \) where \( i \in \{0, 1\} \), the proposal of \( C \) is \( P_C = \{C', C\} \) for all \( C_i \) such that \((C_1, C) \in E^H_1; \omega^H(C', C) \geq \omega^H(C, C)\). Essentially, whenever two segments have maximum global Jaccard distance, this process would forms a larger segment by combining these two.

4 Experiment

4.1 Metrics for performance evaluation

We describe the basic setup for evaluating the proposed approach. We take a data set representing a bipartite graph \((V_0 \cup V_1, E)\), and apply our PSA system to this graph. For any consumer \( v \in V_0 \), we use \( C_v \) to denote the set of commodities that \( v \) is attracted to, i.e., \( C_v = E(v) = \{u \in V_1 \mid \{v, u\} \in E\} \); this set is the desired outcome of the system for \( v \). For the experiments, we fix a percentage \( p \) and “hide” \( p \) percentage of the edges. This results in a subgraph \((V_0 \cup V_1, E')\) where \( E' \subseteq E \). The known set for the consumer \( v \) is \( C^*_v = E'(v) = \{u \in V_1 \mid \{v, u\} \in E'\} \). Similarly, the known set for any commodity \( u \in V_1 \) is \( C^*_u = E'^{-1}(u) = \{v \in V_0 \mid \{v, u\} \in E'\} \). For any \( v \in V_0 \cup V_1 \), the PSA system allocates the known set \( C^*_v \) to the cell corresponding to \( u \).

The PSA system will produce for any consumer \( v \in V_0 \) a segment \( S(v) \subseteq V_0 \) which contains \( v \). We use \( C'_v \) to denote the set of commodities chosen by all members of the segment \( S(v) \), i.e., \( C'_v = \{u \mid \exists v' \in S(v) : \{u, v'\} \in E\} \). Note that, the ideal outcome of the automatic market segmentation mechanism should be the set \( C_v \). To verify the “correctness” of the PSA-based segmentation, we would then need to compare the set \( C'_v \) with the set \( C_v \). The desired outcome is that any consumer \( v \) demonstrates a strong correlation with the commodities in \( C'_v \); this can be analysed using the three metrics: precision, recall and successful deployment rate. Precision and recall are two metric commonly used in data mining for demonstrating the performance of an automatic recommendation system[4].

The precision of user \( v \) is defined as:

\[
\text{precision}(v) = \frac{|C'_v \cap C_v|}{|C'_v|} \tag{5}
\]

Intuitively, the value of precision\((v)\) is the percentage of “correctly identified” commodities within the all “identified” commodities. It captures the amount of commodities chosen by a consumer within the computed segment of this consumer. Naturally, the higher its value, the higher the degree of relevance between \( v \) and the commodity segment \( C'_v \). In particular, the precision of a commodity segmentation should be higher than the its random selection rate \( \frac{|C_v|}{|V_1|} \) ( \( V_1 \) is a set of all commodities), which is the value if \( C'_v \) is chosen randomly.

The recall of \( v \) is defined as:

\[
\text{recall}(v) = \frac{|C'_v \cap C_v|}{|C_v|} \tag{6}
\]
The value of recall($v$) is the percentage of “correctly identified” commodities within the total commodities “chosen by the consumer”. Only considering precision may have some side effects. For example, in order to achieve high precision, a commodity may become conservative to include more commodities. Improving recall with segmentation of a commodity segmentation enforces it to include more attractive commodities. Similarly to precision, the expected recall is $\frac{|C_r^v \cap C_v|}{|C_v \setminus C_r^v|}$. 

The successful deployment rate of $v$ is

$$sdr(v) = \frac{|C_r^v \cap C_v \setminus C_r^v|}{|C_v \setminus C_r^v|}$$

A commodity segmentation should exploit a consumer’s demand by introducing new attractive commodities which are not in a consumer’s preference.

For every consumer segmentation, there must exist at least one commodity segmentation such that each consumer in the segmentation has strong correlation with the commodity segmentation. The great performance of system should find consumer and commodity segmentations which have such correlations as much as possible.

4.2 Data set

We take the Jester data set [3], which contains Internet users’ ratings of a collection of jokes. Although strictly speaking the context provided by this data set is not a market, we may view a user as a consumer and a joke as a commodity, and segmentation of this data set reveals users’ different tastes on humour. We select one of its experimental data sets which contains 24,983 users and 101 jokes. Each user rates at least 36 jokes by giving an integer value ranging from -10 to 10. We further processed the data set by assuming that a user likes a joke if its rating is non-negative. We aim to use PSA system to identify segmentations of people with similar humour taste.

To validate our result, our system should be able to tell that if users like an unknown joke. The data set doesn’t supply additional test data set. So we pick a test data set from the original data set, by keeping for each user, only his rating of 80% of the jokes. Our system processes the test data set. Afterwards, we test how much the PSA system recovers the original rating (on all jokes).

Our system solves the dual segmentations problem. Jester data set is an imbalanced data set because number of consumers is much larger than number of commodities. To understand the performance of our approach on balanced data sets, i.e., those where the number of consumers and number of commodities are both large, we design an experiment which involves a synthetic data set. In generating this synthetic data set, we use the following parameters:

- the number of consumers and commodities ($\sigma_1$)
- the number of segmentations for consumers and commodities ($\sigma_2$)
- the probability of selecting a segmentation in the dual network ($\sigma_3$)
- the probability of generating an edge to a node in a selected segmentation of the dual network ($\sigma_4$)
- the probability of generating an edge to a node in a segmentation which is not selected of the dual network ($\sigma_5$)
If two segmentations on the different sides of a duel network select each other, then the probability of generating edges between nodes in these two segmentations is high. Otherwise the probability should be low. Note that one segmentation has to select at least one segmentation and it also has a probability to select more than one segmentation. Our synthetic data set indeed makes sure that certain relationships are established between segmentations. In the experiment, we give all the edges to our system. Instead, our system needs to find all the segmentations and selected segmentations in the dual network. Our settings for a synthetic data set as follow:

- $\sigma_1 = \sigma_2 = 500$, $\sigma_3 = 0.5$, $\sigma_4 = 0.8$, $\sigma_5 = 0.2$
- $\alpha_0 = \alpha_1 = 0.7$, $\beta_0 = \beta_1 = 0.5$, $\beta_0 = \beta_1 = 0.6$

4.3 Experiment Environment

The PSA framework is designed to be implemented in a distributed environment where cells functions autonomously. Due to physical limitation, we implement the simulation in a single workstation, but allow parallel processes, each simulating a group of cells. In our simulation, each cell only has to process a small amount of local information surrounding it. So, this parallel framework greatly improves computation time. We implement the PSA system using the Java programming language as a multi-threaded application. To improve efficiency, we evenly distribute all the cells in a data set into four parallel threads. Each thread is responsible to execute PSA cells sequentially. So, we simulate a low level parallel computing but it is much efficient in time comparing to a sequential execution of all PSA cells. The detailed specification of our experiment environment is: Windows 7 operating system, Intel Core i5-4570 quad-core CPU 3.2GHz and 16GB RAM.

4.4 Experimental Results and Discussion

We conduct four experiments on the Jester data set. We use PSA system to identify segmentations of users with similar humour taste. In Experiment 1, we take 10 test data sets, consisting of 1000 consumers and 101 commodities each. For any consumer, we retain 80% commodities it links to in the data set. We set $\alpha_0 = 0.35$, $\beta_0 = 0.3$, $\gamma_0 = 0.5$ and $\alpha_1 = 0.45$, $\beta_1 = 0.7$, $\gamma_1 = 0.2$. The results are shown in Figure 1.

![Fig. 1. Results in Experiment 1](image1)

![Fig. 2. Results in Experiment 2](image2)
The Jester data set is imbalanced as the number of users greatly exceeds that of times. So we have to configure different parameters for each dual network. The random selection rate states that, a system has more than 40% chance to correctly determine what to suggest to a consumer if it only makes random guesses. The result shows that our system improves the precision to 80%. In addition, it finds almost 60% of hidden commodities.

In Experiment 2, our goal is to analyse how data set quality influences the results. We take consumers in Dataset 1 from Experiment 1 and generate 10 test data sets by varying the amount of links that are deleted. The high percentage of deleted links, the harder segmentation can be done. We show results of Experiment 2 in Figure 2. Precision describes how less mistakes that a system makes. When we reduce the percentage of given commodities, our system starts to lose capability of finding correlated segmentations within current parameter configuration. Hence, it becomes conservative to make any segmentations but only deploys given commodities. This experiment only shows influences caused by quality of data sets. We keep the parameter configuration from Experiment 1. In practice, we should reconfigure system parameters while dealing with different data sets.

In Experiment 3, we generate 20 test data sets with size from 100 to 2000. We keep the same configuration as in Experiment 1. The results are in Figure 3. The results don’t vary much with respect to to size of the data sets. In Experiment 3, fluctuation of sdr is stronger than other metrics. However, with increasing of the data set size, sdr becomes flat. Prediction of sdr is hard. As shown in Experiment 2, a data set should not include many unknown information otherwise segmentations will be inaccurate. Correlating segmentations in a dual network becomes hard when one side of the dual network has much less segmentations than the other side.

In Experiment 4, we compare our solution with some other existing approaches for market segmentation over the same data sets. The problem is that some approaches such as k-means, x-means and hierarchical clustering require predetermining the number of segmentations. Therefore we only compare our solution to an algorithm which implements a mechanism which can be used to determine number of clusters. Expectation maximization algorithm in WEKA uses cross validation to determine number segmentations in a data set. PSA system determines the number of segmentations on acceptances of different proposals. The results of Experiment 4 are in Figure 4.
EM clustering has high recall and sdr because it deploys all commodities to every consumer segmentation. It is only slightly better if a system randomly guesses the result. The advantage of PSA system is to dynamically segmenting consumers according to commodity segmentation. A proposal captures local information regarding to what requires in a segmentation.

As mentioned above Jester data is imbalanced. We further test our approach in a balanced network where the number of consumers roughly equals to the number of commodities. To make the synthetic data realistic, we includes certain noise that consumers or commodities in a designated segmentation may not have as many connections as others do. Such noise does not affect the experimental results as our system still finds more hidden commodities while keeping a high resolution of precision and recall. Our approach achieves an overall good performance on this synthesised data:

<table>
<thead>
<tr>
<th>precision</th>
<th>recall</th>
<th>sdr</th>
<th>random selection rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.87466</td>
<td>0.939716</td>
<td>0.981968</td>
<td>0.655504</td>
</tr>
</tbody>
</table>

5 Related works

Huang et. al implements a support vector machine to find market segmentations [5]. They conduct experiments on a drink company data set. They show the evidence that support vector machine can outperform better in terms of forming better clusterings than other clustering algorithms such as $k$-means algorithm. Wang takes another clustering algorithm called robust fuzzy clustering in [12]. He improves the original algorithm by pre-processing data set so that noise data can be eliminated. The results show the improvements in terms of quality of clusters as well. Migueis et. al use a variable clustering algorithm to determine consumer segmentations from purchase history dataset in [11]. They assign consumers to each segmentation to illustrate purchasing power of customers. Deng et. al analyses market segmentation in mobile e-commerce of a Japanese chain restaurant in [1]. They propose a hybrid clustering framework consisted of $k$-means algorithm, self-organising feature mapping and particle swarm optimisation. They identified over 70% observed customer segmentations; and each individual approach in the hybrid approach performs less accurate. Kuo et. al investigate different combinations of a set of algorithms in [6]. They conduct experiments on data collected from survey about web user segmentations. They evaluate their results through a business marketing analysis.

6 Conclusion and Future Work

As markets grow and consumers become more sophisticated, the sheer amount and complexity of data makes market analysis increasingly dependent on computer technology such as data mining technique [2]. Currently, most solution models are based on discrete data and lack comparison with data ground truth. In order to face more imminent challenge in this research area, we firstly introduce a new problem formulation: We view market segmentation problem as a
bipartite network, consisted of a consumer network as well as a commodity network. Based on this model, we simulate the formation of segmentations, capturing influence between consumer and commodities. We extend the PSA framework [9, 8] by providing a novel distributed framework for market segmentation. We evaluate our results with ground truth data and receive promising feedback. The challenging part for our current framework is that we have to manually configure our system parameters. Our first extension to this work is to embed a configuration learning mechanism in each data processing unit. We should also develop our system in a real distributed environment. They can identify characteristics of their results from point of views of marketing

References
CORPNET: Towards a Decision Support System for Organizational Network Analysis using Multiplex Interpersonal Relations

Jiamou Liu¹, Anastasia Moskvina¹, and Michael Ouředník²

¹School of Computer and Mathematical Sciences
Auckland University of Technology, New Zealand
{jiamou.liu, anastasia.moskvina}@aut.ac.nz,
²mike.ourednik@gmail.com

Abstract. We introduce CORPNET, a stand-alone application for organizational network analysis (ONA) which provides functions for simulating, analyzing and visualizing the interpersonal relations within corporations. This software tool is different from existing ONA tools in the sense that it is based on a multiplex network model which take into account both formal and informal relations between individuals. We demonstrate that CORPNET can compute individual influence utilizing network centrality, as well as providing useful guidelines regarding stability and leadership styles. Moreover, CORPNET also incorporates a novel benchmark network simulator which generates random informal links within a company. The overall goal is to develop an integrated intelligent decision support system based on organizational networks.

Key words: Organisational network analysis, Bonacich power, formal and informal relations, leadership styles

Availability: CORPNET and its source code can be downloaded from https://github.com/mourednik/corpnet

1 Introduction

Rapid technological and industrial advancement in the last few decades facilitates expansion of companies into giant enterprises. As a company grows, it acquires a larger number of employees, henceforth encompasses increasingly complex interpersonal relations. A major effort in management science is, therefore, to understand how the observed interpersonal relations affect the company performance as a whole. In particular, the emerging field of organizational network analysis (ONA) amounts to a systematic approach to analyze structures within a large organization, thereby providing useful decision supports [4, 5, 7].

We classify interactions among employees into two types: the formal structure, which represents the reporting hierarchy, and the informal structure, which represents the other, more personal connections, such as collaboration, friendship, and advices. As Krackhardt and Hanson pointed out in [11], if the formal
structure is the skeleton of the company, the informal structure is its central nervous system. Ignoring the informal structure could be a costly mistake since much of the work in a company happens despite the formal organization. While the formal structure is static, informal networks are more flexible and adaptive. Hence there is a need for an intelligent decision support system that combines techniques from both static and dynamic network analysis.

There are several software tools created for ONA. The reader is referred to, for examples, the works [10, 14, 15]. These software tools usually perform data visualization, as well as computing common network measures (centrality, network density, cluster analysis etc). However, two significant limitations exist 1) such products are mostly commercially available which made them difficult to be adopted for management science research, and, 2) most importantly, they do not study correlations between formal and informal structures.

In this paper, we introduce CORPNET, a stand-alone software application created to perform flexible organizational network analysis. The underlying organizational model in CORPNET incorporates both formal and informal ties and hence is a multiplex network. The main features of CORPNET include instruments of statistical and stability analysis, community detection, and ability to generate real-life alike hierarchies and social networks. It also provides visualization of the network together with the key indicators above. Using a range of parameters, the software not only allows identification of personal influence in a company, but also reasoning about leadership styles and strategies.

Moreover, CORPNET is also able to generate synthetic benchmark networks, which could be used for ONA research. We point out that existing social network benchmarks are not suitable for organizations with a fixed formal hierarchy. Indeed, the social networks in this context are often based on the homophily principles [12]: people tend to make friends with those who are similar to them. In this spirit, we propose a model that takes into account hierarchical relations. Thus, we enable generating and simulating real world organizational networks within our framework.

The rest of the paper is organized as follows. Section 2 introduces the multiplex organization model. Section 3 discusses Bonacich power, which provides the foundation of individual influences in CORPNET. Section 4 presents main features of the CORPNET software. Section 5 discusses the benchmark networks generated by CORPNET. Section 6 demonstrates how CORPNET can be used to capture leadership styles and Section 7 concludes with remarks on future works.

2 Organizational Network Model of CORPNET

An organizational structure is often defined as a set of positions, groups of positions, reporting relationships and interaction patterns [1]. We use graphs as the basic structure to capture these notions: Nodes in the graph denote the work positions (or employees). There are two types of edges in the graph model:

1. An employee may report to a certain manager. We represent the reporting relation from any employee to her manager by an edge. If an employee does not report to anyone in the company, we refer to her as a top-level manager.
or a root (they could act, for example, as the board of directors). Moreover we also require that all top-level managers are mutually linked by edges. The edges created from the reporting relations and the relation between top-level managers are called strong ties, as they define the most official path of information inside the company. Clearly, the strong ties are directed and hence they form a directed graph.

2. The employees also are associated in a more “random” fashion thanks to different informal links. Such edges are referred to as weak ties as they may undergo frequent changes. Such weak ties are usually mutual between two people and hence we assume the weak ties form an undirected graph.

On the one hand, the strong ties delineate the organizational hierarchy of a firm by featuring reporting relationship. On the other hand, the weak ties enrich the model by including informal relations among employees. More formally we define the following:

**Definition 1 (Organizational network).** An organizational network is a structure $G = (V,R,E_s,E_w)$, where $V$ is a set of nodes, $E_s,E_w \subseteq V^2$ are, respectively, the strong and weak tie relations with the following properties:

1. $R \subseteq V$ is a set of roots and $(r_1,r_2) \in E_s$ for any $r_1 \neq r_2 \in R$: When $R$ only consists of a single node $r$ (i.e., $|R| = 1$), there is an edge $(r,r) \in E_s$ (a self-loop).
2. the pair $(V,E_s)$ forms a directed acyclic graph, where every node apart from nodes in $R$ has an incoming edge from another node;
3. the pair $(V,E_w)$ forms an undirected graph.

To refine this network model, we put forward constraints that reflect important principles in real-life organizational networks. In particular we capture the following principles:

1. **One Manager.** If a person has several sources of instructions, a head-on collision may occur which results in confusion and conflicts. Hence in an effective organization one would require every employee to report to at most one manager. Thus the strong tie relation on the subordinates below a certain root will be shaped as a tree.

**Principle 1 (One Manager):** Each node has exactly one incoming directed edge, which represents relationship with its manager, i.e., for all $u \in V$ there is a unique $v \in V$ with $(v,u) \in E_s$.

2. **Limited Capacity.** Capacity is an important concept in social network analysis which asserts that anyone can maintain only a limited number of interpersonal relations, due to limited time and effort [3]. For example, the number of strong personal links for networks of more than 500 nodes on Facebook varies from 10 to 20 [6].

Let $\Delta$ be an abstract quantity that defines the maximum amount of resources (working hours, for instance) that a person can distribute between his or her ties. We assume that a person needs $s$ resources and $w$ resources to maintain
a strong and a weak tie, respectively. Therefore, for any node \( v \), if \( E_s(v) \) is the number of directed edges (including self-loop), and \( E_w(v) \) is the number of undirected edges, then this person spends \( E_s(v) \times s + E_w(v) \times w \leq \Delta \) resources to maintain all his connections.

Call the ratio \( \delta = \frac{w}{s} \) the correlation coefficient. We define the relative degree of a node \( v \in V \) as \( d(v) = E_s(v) + E_w(v) \times \delta \), where \( E_s(v), E_w(v) \) are the numbers of strong ties (including both incoming and outgoing edges) and weak ties \( v \) maintains, respectively.

**Principle 2 (Maximal Relative Capacity):** There is a constant \( c \) such that for all \( v \in V \), the relative degree \( d(v) \leq c \).

3. **Limited Height** In management science, *flattening* or *delayering* refers to the process of elimination of layers in a firm’s organizational hierarchy [20]. Back in 1950s companies had up to twenty layers of hierarchy while by the end of the twentieth century they were trimmed to five or six. This phenomenon has been thoroughly investigated in the literature [18,19]. Similarly to the “6-degree of separation” in social networks, in an effective organization, the top managers are usually not very far from all subordinates. Then, the third principle is simply as following:

**Principle 3 (Limited Height):** The largest level of hierarchy is not more than some given number \( h \).

### 3 Individual Influences in an Organizational Network

A crucial functionality of CORPNET is the automatic identification of individual influences in an organization. Similarly to the concept of social capitals (see for example [6]), we define a notion of “positional advantage”. Clearly, a position in a company is more advantageous if it is closer to the top managers. Moreover, a large number of personal connections provide more sources of information. Finally, the span of control indicates the number of subordinates to a manager, hence their level of involvement in making decisions within the network [9]. Based on this, we assign a weight between the values 0 and 1 to all edges \( e \). The weight function \( \mu \) depends on two parameters \( \rho \) and \( k \):

- **Strong ties:** If \( e \in E_s \), we assign \( \mu_{\rho,k}(e) = 1 \) if any end point of \( e \) is a non-root; and \( \mu_{\rho,k}(e) = \rho \) otherwise. The edges between two roots (including the self-loop when there is only a single root) are special. A weight \( \rho \) of 1 suggests that it has the same affect as all the other strong ties, while 0 means that these edges only affects the capacity of the root node but not the influence. We will show later that the weight \( \rho \) is important for defining management styles: a larger \( \rho \) indicates a more “autocratic” style of management.
- **Weak ties:** If \( e \in E_w \), we assign \( \mu_{\rho,k}(e) = k \) where \( k \) is a parameter to the model. The range of \( k \) guarantees that directed edges are more important than undirected. An edge from \( A \) to \( B \) can be interpreted as \( A \) has influence over \( B \) ranked as the weight of this edge.
- **Back flow**: It is natural to assume that the influence between an employee and her manager is not one-way: while the manager influences the employee through a strong tie, the employee also influences her manager through social interaction, and hence can be regarded as a weak tie, i.e., a back flow. Hence we set $\mu_{p,k}(e) = k$ where $e = (u, v)$ whenever $u$ reports to $v$.

Based on the definition above, we regard any organizational network as a *weighted influence graph*; an example of this is shown in Figure 1.

**Definition 2 (Weight influence graph).** Let $G = (V, R, E_s, E_w)$ be an organizational network. The weighted influence graph of $G$ is

$$W(G) = (V, R, E_s, E_w, k, \rho, \mu_{p,k})$$

where $k, \rho$ and the weight function $\mu_{p,k}$ are defined as above.

![Fig. 1. An organizational network (left) and a weighted influence graph (right)](image)

To capture individual influences in an organizational network, we use a centrality notion that is adapted from **Bonacich power**. Philip Bonacich in [2] expanded the eigenvalue centrality by introducing two additional parameters $\alpha$ and $\beta$. Let $A$ be an adjacency matrix of a network. Then for any node $i$, the Bonacich power is defined as follows:

$$p_i(\alpha, \beta) = \sum_j (\alpha + \beta p_j) A_{ij}$$

(3.1)

We collect Bonacich power of all nodes in a vector $p(\alpha, \beta)$:

$$p(\alpha, \beta) = \alpha (I - \beta A)^{-1} Ae,$$

(3.2)

where $I$ is an identity matrix, and $e$ is a column vector of ones. The parameter $\alpha$ is a scalar that affects only the length of the power vector $p$. It means that we use it only to normalize the powers. In this paper $\alpha$ is selected such that the squared length of $p$ equals the number of nodes in the network. Then, $p_i(\alpha, \beta) = 1$ means (approximately) that position $i$ does not have an unusually large or small degree of centrality.

Recall from Principle 3 that $c$ denotes the upperbound of relative degrees of individuals in the network. We use $\lambda$ to denote the dominating eigenvalue of the adjacency matrix of the weighted influence graph. The parameter $\beta$ can be any value on the interval $[-\frac{1}{\lambda}, \frac{1}{\lambda}]$ where $\lambda$ is the largest eigenvalue of the
weighted influence graph. When $\beta = 0$, Bonacich power coincides with degree centrality. When $\beta > 0$, a node become more powerful as its neighbors become more powerful. In contrast, when $\beta$ is negative, nodes become more powerful as their neighbors become less powerful (this is useful to capture negative exchange networks). Intuitively, we use the parameter $0 < \beta < 1$ to characterize the loss of control of the managers [9]. The more connections an individual has, the less their power depends on each of their neighbors’ powers. Capacity indirectly indicates how much time a worker spends with each of their subordinates, friends or collaborators. Therefore, we assume that $\beta$ is inversely proportional to the capacity minus one (one is for the relation with the manager):

$$\beta < \begin{cases} \frac{1}{\lambda} & \text{if } \lambda > 1, \\ \frac{1}{c-1} & \text{otherwise} \end{cases} \quad (3.3)$$

4 Main Features of CORPNET

CORPNET is a software application for interactive simulation, analysis and visualization based on the network model above. It is developed using the Scala programming language, and runs on the Java Virtual Machine. The program allows user customization over a range of key parameters ($\beta$, $\Delta$, $w$, $k$, $\rho$, $h$ as discussed above).

Fig. 2. CORPNET user interface: a tree layout (left) and a force directed layout with a power distribution (right)

The user interface consists of three parts; a menu bar, a visualization area, and log output area. The internal architecture is multithreaded; all user interface actions trigger an asynchronous message passed to a finite state machine, which in turn executes computations on separate threads. The numerical calculations on the network graph adjacency matrices are handled by Breeze, a numerical
processing library for Scala, which makes use of native BLAS, LAPACK and ARPACK libraries, thus achieving maximum performance.

The network graph can be visualized as either a tree layout or a force directed layout. The tree layout is hierarchical with respect to the directed edges only. The color of a node can represent either its power, or its membership within a detected community. See Fig. 2 for examples of both layouts.

CORPNET computes Bonacich power as defined above and visualize this information in multiple ways. For both layout styles, a darker shade of blue indicates a higher relative power within the network. The tree layout also displays the power of a node as numeric text. The force layout draws nodes with varying sizes, such that more powerful nodes are relatively larger. By default, the power values are immediately recalculated for all nodes after the user modifies the network by either adding/removing nodes and edges, or adjusting model parameters. This automatic recalculation can be disabled, which is convenient when making a batch of changes to a large network (it can take a few seconds to recalculate powers with more than a thousand nodes). CORPNET is able to perform ONA from the following perspectives:

**Stability Analysis.** We posit that the individual powers influence significantly organizational stability. Intuitively, power of individuals who are closer to the top managers should be greater than the power of individuals further down the hierarchy. This led us to the idea of the stable and unstable networks: an employee in some hierarchy layer \( l \) becomes unsatisfied if there is another employee whose power and the hierarchy level are greater. A node is considered strongly (autocratically) unstable if it has lower power than at least one node on a lower level. A node is considered weakly unstable if it has lower power than at least one of its descendants. A level is considered strongly unstable if it contains a strongly unstable node.

The program automatically analyzes the stability of the network after calculating the Bonacich power. The analysis result is displayed in the log output area. It shows whether the network is strongly stable, weakly stable. If the network contains any unstable nodes, the report will show the degree of each type of instability, as the ratio of the total number of unstable nodes to the total number of nodes.

**Community Detection.** CORPNET incorporates a module which computes communities in the organizations based on strong and weak ties using Newman’s spectral algorithm [8]. This is visualized by node colors which represent communities detected.

**Power Distribution Plots** Two plots are available providing statistical information regarding individual influences:

- Descending power grouped by level: A scatter plot of individual node powers. The nodes are arranged from left to right in descending order of power, grouped by level such that the nodes on higher levels are to the left of nodes on lower levels.
– **Power histogram**: A histogram of node powers with a configurable number of bins.

![Power histogram](image)

**Fig. 3.** Power distribution plots: Descending power grouped by level (left) and Power histogram (right)

## 5 Organization Structure Simulation Using CORPNET

To facilitate experiments on organizational networks, CORPNET incorporates a network simulator which is able to generate synthetic benchmark organizational networks.

**Random tree generator.** This module generates a tree with a random number of children per node. The number of children is sampled from a normal distribution with a specified mean and standard deviation. The mean of the normal distribution can be incremented for each level. For example, with *initial mean* 3.0, and *mean increment* 0.5, the first level will have mean 3.0, the second level will have mean 3.5, and so on. If the standard deviation and mean increment are both set to zero, then the result is a perfect tree, with exactly the *initial mean* number of children for all nodes.

**Social network generator.** This module generates a benchmark social network of undirected edges over the existing hierarchical network. Here, typical benchmark graphs in social network analysis is not appropriate: numerous management theories argue from a homophily point of view and assert that employees are more likely to establish social connections within a certain “circle”, such as departments, offices. Moreover, individuals tend to establish personal ties with those that are at the same or similar levels in the hierarchy [12, 17]. No social network model so far has been defined taking into account this multiplex view of an organization (Existing benchmark graphs such as planted $\ell$-partition, relaxed caveman graphs and the LFR graphs [8] are not based on a multiplex model of strong and weak ties). Hence we need to provide our own benchmark graph.

We adopt a distributed approach where each node randomly chooses to set up weak-ties with other, in such a way that closer nodes (in distance) enjoy
a higher “probability” of a weak tie. The result is a random network that not only captures the main characteristics of social networks (such as community structure), but also entails the reporting hierarchy of the network; see Fig. 4 for a generated network visualized using a force-directed method. The community structure clearly resembles departments and reflect hierarchical levels in an organization.

In Fig. 5, we consider a hierarchy with no social network on it. In this case, we can see that clusters reflect departments in some sense, but not the levels of hierarchy. However, when we enrich this hierarchy with a generated social network, we get a quite different picture as in Fig. 5. The modularity detection algorithms applied to randomly generated trees and social networks as explained above demonstrate that two our propositions hold (see Fig. 6): (1) Clusters typically reflect department: people in the same department tend to form a cluster. (2) Clusters typically reveal levels: managers in the same level tend to form a cluster.

6 Analyzing Leadership Styles Using CORPNET

A important question in management studies looks at how performance relies on leadership styles in a company [13]. We suggest that CORPNET allows a more
rigorous analysis of the connections between personal ties and leadership styles. Management styles in a company are classified by the level of control exercised by the top managers. For example, managers in an autocratic company make decisions unilaterally with no initiatives from the bottom while in a democratic company, decisions are made by majority rather than by the top managers. The following are main leadership styles adopted by companies and widely studied [16]. Our multiplex network model allows us to capture them in a formal way:

**Autocratic style.** Here the decisions are made from the top management unilaterally. There are few or no initiatives from the bottom of the hierarchy. The number of connections of any individual employee is relatively low because maintaining a reporting relation requires a large amount of resources. Hence we say that an organizational network (with a fixed weighted influence graph) is autocratic if $\rho > 1$, $k$ is very small (i.e. within the range $[0, 0.1]$), and the resources required to maintain weak ties is small $(w \in (0, 0.5))$. The benefit of this style is high stability, while the negative effect is the lack of motivation of employees.

**Democratic style.** Here the decisions are made by majority of the employees. There are many initiatives from the bottom of the hierarchy and collaboration requires as much resources to maintain as the reporting relations. Thus we say that an organizational network is democratic if $\rho = 0$, $k \in [0.5, 1)$ and $w$ is close to 1. The benefit of this style lies in job satisfaction and quality, however it does mean a higher level of instability and inefficiency.

**Paternalistic (consultative) style.** This leadership style sits somewhere between autocratic and democratic. While the decisions are made mainly from the top managers, they take into account the best interests of the employees. Influence is mainly one-directional (downwards), but feedback also are encouraged. Hence we define an organizational network to be paternalistic if $\rho > 0$, $k \in (0.1, 0.5)$, $w \in [0.5, 1)$.

**Chaotic style.** This is a more recent style of management, which gives employees total control over decision making. Here weak ties has become the dominant personal links and thus require a larger weight in their influence and a large amount of resources to maintain. We define an organizational network to be chaotic if $\rho = 0$, $k = 1$ and $w \geq 1$. One would expect that any chaotic organizational structure to be unstable.
Figure 6 shows three typical networks with different management styles: democratic, paternalistic, and democratic. One can clearly identify that the distribution of power in such networks is quite different, and reflect the analysis above. We run several experiments by generating different organizational networks with social networks, and get the following results in regards to stability of the networks:

7 Conclusion and Future Work

The current version of CORPNET is a prototype of an ONA tool that performs integrated functionalities of simulation, analysis and visualization. The software incorporates several novel ideas in ONA that include a multiplex network model, and a new benchmark model used to generate informal ties in a corporation. We show how the tool can be use to compute a range of organizational information such as stability and leadership style. Our overall goal is to develop an intelligent
decision support system for corporations based on organizational networks which provides useful managerial and strategic advices.

Future work here include enriching the functionalities of the software to more aspects of organization networks. While the current model allows one to discover individual influence, it would be useful to develop a measure of individual effectiveness and performance and investigate how it affects the performance of the overall company. On the other hand, we would also like to investigate detecting “hidden” hierarchies that are not revealed by the strong ties.

References

15. SYNDIO. Retrieved from https://synd.io/
Author Index

Adamatti, Diana Francisca 2
Bai, Quan 1, 7
Crosscombe, Michael 5
Dimuro, Graçaliz 2
Ito, Takayuki 4
Khalid, Nor 1
Kiekintveld, Christopher 6
Lawry, Jonathan 5
Liu, Jiakun 3
Liu, Jiamou 7, 8
Moskvina, Anastasia 8
Otsuka, Takanobu 4
Ourednik, Michael 8
Ren, Fenghui 3
Rojas, Yunevda 2
Torii, Yoshitaka 4
Veliz, Oscar 6
Wei, Ziheng 7
Zhang, Jihang 3
Zhang, Minjie 3
## Keyword Index

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive forwarder selection</td>
<td>1</td>
</tr>
<tr>
<td>Agent Based Modelling</td>
<td>6</td>
</tr>
<tr>
<td>Agent Based Simulation</td>
<td>6</td>
</tr>
<tr>
<td>Agent-Based Modelling</td>
<td>5</td>
</tr>
<tr>
<td>Belief Aggregation</td>
<td>5</td>
</tr>
<tr>
<td>Bonacich power</td>
<td>8</td>
</tr>
<tr>
<td>Complex System</td>
<td>6</td>
</tr>
<tr>
<td>Consensus</td>
<td>5</td>
</tr>
<tr>
<td>Dependence Relations</td>
<td>2</td>
</tr>
<tr>
<td>Disaster Informatics</td>
<td>4</td>
</tr>
<tr>
<td>Distributed Wireless Sensor Network</td>
<td>1</td>
</tr>
<tr>
<td>Domain Transportation</td>
<td>3</td>
</tr>
<tr>
<td>Emergency management</td>
<td>3</td>
</tr>
<tr>
<td>Field Informatics</td>
<td>4</td>
</tr>
<tr>
<td>formal and informal relations</td>
<td>8</td>
</tr>
<tr>
<td>Game Theory</td>
<td>6</td>
</tr>
<tr>
<td>leadership styles</td>
<td>8</td>
</tr>
<tr>
<td>Many-Valued Logics</td>
<td>5</td>
</tr>
<tr>
<td>Market segmentation</td>
<td>7</td>
</tr>
<tr>
<td>Multiagent system</td>
<td>3</td>
</tr>
<tr>
<td>multiple parameters</td>
<td>1</td>
</tr>
<tr>
<td>Organizational network analysis</td>
<td>8</td>
</tr>
<tr>
<td>PSA framework</td>
<td>7</td>
</tr>
<tr>
<td>Q-learning</td>
<td>1</td>
</tr>
<tr>
<td>Reputation</td>
<td>2</td>
</tr>
<tr>
<td>Resource allocation</td>
<td>3</td>
</tr>
<tr>
<td>simulation</td>
<td>7</td>
</tr>
<tr>
<td>Social Exchanges</td>
<td>2</td>
</tr>
<tr>
<td>Trust Transference</td>
<td>2</td>
</tr>
</tbody>
</table>
Wireless Sensor Network 4