Christian Bettstetter
Carlos Gershenson (Eds.)

LNCS 6557

Self-Organizing Systems

5th International Workshop, IWSOS 2011
Karlsruhe, Germany, February 2011
Proceedings

Springer

Lecture Notes in Computer Science 6557

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Christian Bettstetter Carlos Gershenson (Eds.)

Self-Organizing Systems

5th International Workshop, IWSOS 2011
Karlsruhe, Germany, February 23-24, 2011
Proceedings

Springer

Volume Editors

Christian Bettstetter
University of Klagenfurt
Mobile Systems Group, Institute of Networked and Embedded Systems
9020 Klagenfurt, Austria
and
Lakeside Labs GmbH, Lakeside B04, 9020 Klagenfurt, Austria
E-mail: christian.bettstetter@uni-klu.ac.at

Carlos Gershenson
Universidad Nacional Autónoma de México
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
Ciudad Universitaria, A.P. 20-726, 01000 México, D.F., México
E-mail: cgg@unam.mx

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-19166-4 e-ISBN 978-3-642-19167-1
DOI 10.1007/978-3-642-19167-1
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011920515

CR Subject Classification (1998): C.2, D.4.4, C.4, H.3, H.2.8

LNCS Sublibrary: SL 5 – Computer Communication Networks and Telecommunications

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This book contains research articles from the International Workshop on Self-Organizing Systems (IWSOS), held in Karlsruhe, Germany, in February 2011. This was the fifth workshop in a series of multidisciplinary events dedicated to self-organization in networked systems with the main focus on communication and computer networks.

The concept of self-organization is becoming increasingly popular in various branches of technology. A self-organizing system may be characterized by global, coordinated activity arising spontaneously from local interactions between the system's components. This activity is distributed over all components, without a central controller supervising or directing the behavior. Self-organization relates the behavior of the individual components (the microscopic level) to the resulting structure and functionality of the overall system (the macroscopic level). Simple interactions at the microscopic level may give rise to complex, adaptive, and robust behavior at the macroscopic level.

The necessity of self-organization in technological networks is caused by the growing scale, complexity, and dynamics of future networked systems. This is because traditional methods tend to be reductionistic, i.e., they neglect the effect of interactions between components. However, in complex networked systems, interactions cannot be ignored, since they are relevant for the future state of the system. In this sense, self-organization becomes a useful approach for dealing with the complexity inherent in networked systems.

The workshop series brings together leading international researchers to create a visionary forum for discussing the future of self-organization. It addresses theoretical aspects of self-organization as well as applications in communication and computer networks and robot networks.

The IWSOS 2011 committee received 25 submissions from 16 countries. Two papers were immediately rejected due to incomplete author information. Each of the 23 submissions found to be eligible was reviewed by at least three members of the technical program committee. In total, 83 reviews were performed. Based on these reviews, nine papers were accepted. Five of these papers underwent a shepherding process by one of the TPC members to ensure revisions for the final version. One additional publication was invited for the workshop. The authors of the papers are from Germany, Finland, Canada, Belgium, Austria, Greece, Italy, and Japan.

Key topics include the following:

- Design and analysis of self-organizing and self-managing systems
- Techniques and tools for modeling self-organizing systems
- Robustness and adaptation in self-organizing systems, including self-protection, diagnosis, and healing
- Self-configuration and self-optimization

- Self-organizing group and pattern formation
- Self-organizing synchronization
- Self-organizing resource allocation
- Self-organizing mechanisms for task allocation and coordination
- Structure and dynamics of self-organizing networks
- Applications of self-organizing networks and networked systems
- Peer-to-peer networks, vehicular networks, zeroconfiguration protocols

The workshop also features two keynote talks given by Hermann Haken (Stuttgart University) and Hod Lipson (Cornell University) as well as a poster session with a student research competition.

January 2011

Christian Bettstetter
Carlos Gershenson

Organization

IWSOS 2011 was organized by the Institute of Telematics, Karlsruhe Institute of Technology (KIT).

General Chairs

Martina Zitterbart	KIT, Germany
Hermann de Meer	Univ. Passau, Germany

Program Chairs

Christian Bettstetter	Univ. Klagenfurt and Lakeside Labs, Austria
Carlos Gershenson	Univ. Nacional Autónoma de México, Mexico

Local Organization

Christoph P. Mayer	KIT, Germany
Denis Martin	KIT, Germany

Steering Committee

Hermann de Meer	Univ. Passau, Germany
David Hutchison	Lancaster University, UK
Bernhard Plattner	ETH Zurich, Switzerland
James Sterbenz	University of Kansas, USA
Randy Katz	UC Berkeley, USA
Georg Carle	TU Munich, Germany
Karin Anna Hummel	Univ. Vienna, Austria

Program Committee

Karl Aberer	EPFL, Switzerland
Christos Ampatzis	European Research Council, Belgium
Ozalp Babaoglu	Univ. Bologna, Italy
Sandford Bessler	FTW Vienna, Austria
Yuriy Brun	Univ. Washington, USA
Sonja Buchegger	KTH Stockholm, Sweden
Iacopo Carreras	Create-Net, Italy
Alexander Clemm	Cisco Systems, USA
Giovanna Di Marzo Serugendo	Univ. London, UK

Falko Dressler	Univ. Erlangen, Germany
Wilfried Elmenreich	Univ. Klagenfurt, Austria
Salima Hassas	Univ. Claude Bernard-Lyon 1, France
Francis Heylighen	Free Univ. Brussels, Belgium
Tom Holvoet	Katholieke Univ. Leuven, Belgium
Karin Anna Hummel	Univ. Vienna, Austria
Geoffrey James	CSIRO ICT Centre, Australia
Mark Jelasity	University of Szeged, Hungary
Holger Karl	Univ. Paderborn, Germany
Alexander Keller	IBM Research, USA
Rajesh Krishnan	Science and Technology Associates, USA
Nick Maxemchuk	Columbia University, USA
Van Parunak	NewVectors, USA
Christian Prehofer	Fraunhofer, Germany
Susana Sargento	Univ. Aveiro, Portugal
Hiroki Sayama	Univ. Binghamton, USA
Marcus Schoeller	NEC Laboratories Europe, Germany
Sidi-Mohammed Senouci	Orange Labs, France
Paul Smith	Univ. Lancaster, UK
Martha Steenstrup	Stow Research, USA
Martin Stetter	Hochschule Weihenstephan-Triesdorf, Germany
John Strassner	Univ. Pohang, Korea
Alexander Tyrrell	DOCOMO Euro-Labs, Germany
Patrick Wüchner	Univ. Passau, Germany

Sponsoring Institutions

Wiley, UK
Lakeside Labs

Table of Contents

Methods for Approximations of Quantitative Measures in Self-Organizing Systems

Richard Holzer and Hermann de Meer

Faculty of Informatics and Mathematics, University of Passau, Innstrasse 43, 94032 Passau, Germany
{holzer,demeer}@fim.uni-passau.de

Abstract. For analyzing properties of complex systems, a mathematical model for these systems is useful. In micro-level modeling a multigraph can be used to describe the connections between objects. The behavior of the objects in the system can be described by (stochastic) automatons. In such a model, quantitative measures can be defined for the analysis of the systems or for the design of new systems. Due to the high complexity, it is usually impossible to calculate the exact values of the measures, so approximation methods are needed. In this paper we investigate some approximation methods to be able to calculate quantitative measures in a micro-level model of a complex system. To analyze the practical usability of the concepts, the methods are applied to a slot synchronization algorithm in wireless sensor networks.

Keywords: Self-Organziation, Mathematical modeling, Systems, Quantitative measures, Approximation.

1 Introduction

Self-organizing systems provide mechanisms to manage themselves as much as possible to reduce administrative requirements for users and operators. Since such systems are usually very complex, mathematical models are used for the analysis of the systems and for the design of new systems. In micro-level modeling the behavior of each entity of the system and the communication between the entities are described. Macro-level modeling uses the technique of aggregation to derive a model for the system variables of interest. Each macro-state can be seen as an equivalence class of micro-states. Quantitative measures [1] [2] can be used as a link from micro-level modeling to macro-level modeling. These measures are defined in a micro-level model and they measure a global property of the system. The dynamic change of the values of the measures during the time yields a macro-level model. These measures allow the analysis of the existing systems with respect to self-organizing properties like autonomy or emergence

[1] This research is partially supported by the SOCIONICAL project (IP, FP7 Call 3, ICT-2007-3-231288), by the ResumeNet project (STREP, FP7 Call 2, ICT-2007-2-224619) and by the Network of Excellence EuroNF (IST, FP7, ICT-2007-1-216366).

C. Bettstetter and C. Gershenson (Eds.): IWSOS 2011, LNCS 6557, pp. 1–15, 2011.

and to optimize some system parameters with respect to a given goal. Also for the design of new systems the measures can help to compare different rule sets or different system parameters for optimization. Further discussions about the importance of quantitative measures can be found in [1], [2]. A practical example, where a quantitative measure is used for an optimization of system parameters in an intrusion detection system can be found in [3].

Unfortunately, many systems are too complex to be able to calculate the exact values of the quantitative measures, because the measures usually consider the global state space, which grows exponentially with the number of entities.

This paper investigates some approximation methods for the calculation of quantitative measures. Section 2 gives an overview of the related work and Section 3 recalls the micro-level model of [2] for complex systems. In Section 4 some approximation methods for quantitative measures are explained. In Section 5 we investigate, how the approximation methods can be used for the calculation of the quantitative measures for emergence, target orientation and resilience. In Section 6, we apply the approximation methods to the model of slot synchronization in wireless networks, which was defined in [1], [2]. Section 7 contains some discussions and advanced conclusions of the methods described in this paper.

2 Related Work

In the last years, much research has been done in the field of self-organizing systems. The main properties of self-organizing systems [4] [5] are self-maintenance, adaptivity, autonomy, emergence, decentralization, and optimization. A non-technical overview of self-organization can be found in [5]. Other definitions and properties of self-organizing systems can be found in information theory [6], thermodynamics [7], cybernetics [8], [9], [10] and synergetics [11]. [12] gives a description of the design of self-organizing systems. [13] gives a systematic overview on micro-level and macro-level modeling formalisms suitable for modeling self-organizing systems. A survey about practical applications of self-organization can be found in [14]. For modeling continuous self-organizing systems and a comparison between discrete and continuous modeling see [15]. Quantitative measures of autonomy, emergence, adaptivity, target orientation, homogeneity, resilience and global-state awareness, can be found in [1], [2], [13], [16] and [3]. Other approaches to quantitative emergence are given in [16] and [17]. In [16] emergence is defined as an entropy difference by considering the change of order within the system. [17] analyses and improves this approach by introducing a multivariate entropy measure for continuous variables, and by using different divergence measures for the comparison of the corresponding density functions. The Parzen window approach, which is used in this paper, was also used in [17] as an estimation method for the density functions.

In this paper we use the definition of emergence of [1], which measures dependencies between communications in the system. The methods of [1], [2] for discrete micro-level modeling with stochastic automatons are also used in this paper. The major drawback of the measures defined in [1], [2] was, that the

state space in practical applications is usually too large to be able to calculate the measures analytically. This paper contributes to this problem: We analyze different approximation methods with respect to usability for the approximation of quantitative measures in complex systems.

3 Discrete Micro-level Model

For modeling discrete systems, we use the methods of [2], which are based on the ideas of [5]: The topology of a system is represented by a multigraph G, where V is the set of vertices and K is the set of directed edges. The behavior of each entity $v \in V$ is described by a stochastic automaton a_v, which has a set of internal states S_v and uses an alphabet A of symbols to communicate with the automatons of other nodes by sending a symbol through an edge $k \in K$ to a successor node. Since the automaton does not need to be deterministic, probability distributions are used to describe the outputs and the state transition function of each stochastic automaton. The influence of the environment on the system can be modeled by special vertices (external nodes) in the multigraph [2]. With these concepts micro-level models can be built for a wide variety of complex systems of the real world, e.g. systems that appear in biology, physics, computer science or any other field. Assume that we would like to analyze a system, e.g. a computer network. Then each node of the network corresponds to a vertex of the multigraph. If one node of the network is able to communicate with another node, then we draw an edge between the vertices in the graph. The behavior of each node is modeled by a stochastic automaton, which describes, how the internal state changes for each input, which it gets from the other nodes.

When we consider the global view on the system at a point of time, then we see a current local state inside each automaton and a current value on each edge, which is transmitted from one node to another node. Such a global view is called *configuration*. It represents a snapshot of the system. For a configuration c and a set $T \subseteq K$ of edges the assignment of the edges in T in the configuration is also denoted by $c|_T$.

To analyze the behavior of a system, we initialize it at time $t_0 = 0$ by choosing a start configuration $c_0 \in \Gamma$, where Γ is the set of all possible start configurations and $P_\Gamma(c)$ is the probability, that $c \in \Gamma$ is used for the initialization of the system. Then the automatons produce a sequence $c_0 \to c_1 \to c_2 \to \dots$ of configurations during the run of the system. Since the automata and the initialization are not deterministic, the sequence $c_0 \to c_1 \to c_2 \to \dots$ is not uniquely determined by the system, but it depends on random events. So for each time $t \geq 0$, we have a random variable Conf_t, which describes, with which probability $P(\mathrm{Conf}_t = c)$ the system is in a given configuration c at time t.

For measuring the information in a system we use the statistical entropy: For a discrete random variable X taking values from a set W the *entropy* $H(X)$ *of* X is defined by [18]

$$H(X) = -\sum_{w \in W} P(X = w) \log_2 P(X = w).$$

The entropy measures, how many bits are needed to encode the outcome of the random variable in an optimal way. For example, the entropy $H(Conf_t)$ measures the amount of information contained in the configuration of the system at time t. The entropy $H(Conf_t|_K)$ measures the amount of information, which is communicated between the entities at time t.

4 Methods for Approximations

For the analysis of global properties of the system, quantitative measures can be used. Unfortunately, entropies of global random variables like $H(Conf_t)$ are difficult to calculate analytically, because complex systems have a huge global state space: The set of all configurations grows exponentially with the number of the entities in the system. Therefore we need methods to approximate the needed values. We can use simulation runs to be able to approximate probabilities and entropies. Let R be the number of simulation runs. Each simulation run leads to a time series, which is the configuration sequence $c_0 \to c_1 \to c_2 \to \dots$ produced by the simulation run. Then the probability of a value $a \in A$ on a single edge $k \in K$ at time $t \geq 0$ can be approximated by the relative frequency $rel_{t,k,a}$, which is the number of time series, which contains the value a on the edge k at time t, divided by R:

$$P(Conf_t|_{\{k\}} = a) \approx rel_{t,k,a} := \frac{1}{R} \cdot r_{t,k,a}$$
$$r_{t,k,a} := |\{s \ : \ s \text{ is a time series with } Conf_t|_{\{k\}} = a\}|$$

Unfortunately, this method is still too complex for probabilities of global valuations like $P(Conf_t = c)$ or $P(Conf_t|_K = \underline{a})$ for $\underline{a} \in A^K$. The range of values for these random variables is too large, so the relative frequencies received from simulation runs are too inaccurate for the approximation of probabilities. For such global probabilities, we investigate three different approximation methods:

1. Classification
   The probability space is divided into different classes. The relative frequency for each class is calculated by considering the time series. The probability for a single element of the class can be approximated by the relative frequency of the class divided by the size of the class. For $P(Conf_t|_K = \underline{a})$ the classification can be done by choosing a subset $K_0 \subseteq K$ and build $|A|^{|K_0|}$ equivalence classes $[b] := \{a \in A^K \ : \ a|_{K_0} = b\}$ for $b \in A^{K_0}$, where the size of each class is $|[b]| = |A|^{|K|-|K_0|}$. Now the relative frequencies $rel_{t,K_0,[b]}$ are calculated for each class $[b]$ and the probability $P(Conf_t|_K = \underline{a})$ is approximated by $P(Conf_t|_K = \underline{a}) \approx \frac{1}{|[b]|} \cdot rel_{t,K_0,[b]}$ for $b = \underline{a}|_{K_0}$. Analogously this concept can be used for the approximation of $P(Conf_t = c)$: In this case we do not only consider the valuations $c|_K$ of edges but also the internal states of the nodes. The classification can be done by choosing a subset $V_0 \subseteq V$ of nodes and a subset $K_0 \subseteq K$ of edges. Then each class is characterized by a valuation $b : K_0 \to A$ of K_0 and the states $g \in \prod_{v \in V_0} S_v$ of the nodes

$v \in V_0$. The number of equivalence classes is $A^{K_0} \cdot \prod_{v \in V_0} |S_v|$. For example, if we have $A = \{0, 1\}$ and a single edge $K_0 = \{k\}$ and $V_0 = \emptyset$, then we have two equivalence classes of configurations: In the first class are all configurations c with $c_K(k) = 0$, and in the second class all configurations have the value 1 on k. After calculating the approximations of the probabilities, we get an approximation of the entropies. For the entropy $H(\mathrm{Conf}_t\,|_K)$ of the edge valuations we have

$$\begin{aligned} H(\mathrm{Conf}_t\,|_K) &= -\sum_{\underline{a} \in A^K} P(\mathrm{Conf}_t\,|_K = \underline{a}) \log_2 P(\mathrm{Conf}_t\,|_K = \underline{a}) \\ &= -\sum_{[b]\text{class}} \sum_{\underline{a} \in [b]} P(\mathrm{Conf}_t\,|_K = \underline{a}) \log_2 P(\mathrm{Conf}_t\,|_K = \underline{a}) \\ &\approx -\sum_{[b]\text{class}} \sum_{\underline{a} \in [b]} \frac{1}{|[b]|} \cdot rel_{t,K_0,[b]} \log_2 \left(\frac{1}{|[b]|} \cdot rel_{t,K_0,[b]} \right) \\ &= -\sum_{[b]\text{class}} rel_{t,K_0,[b]} \log_2 \left(\frac{1}{|[b]|} \cdot rel_{t,K_0,[b]} \right) \end{aligned}$$

A similar formula can be obtained for the approximation of the entropy $H(Conf_t)$, where the classes are defined not only by edge valuations but also by some internal states of some nodes. Note that this approximation method for $H(\mathrm{Conf}_t\,|_K)$ and $H(\mathrm{Conf}_t)$ is also efficient for a large number of classes (low size of each class $[b]$), since only the summands with $rel_{t,K_0,[b]} \neq 0$ have to be taken into account.

2. Parzen window approach
For a random variable X with a sample set $W = \{w_1, \ldots, w_R\} \subseteq \mathbb{R}^{dim}$ (observations of X) we can use the kernel density estimator based on a Gaussian kernel [17], [19]

$$p(a) = \frac{1}{R} \sum_{j=1}^{R} \frac{1}{(2\pi h^2)^{dim/2}} \exp(-\frac{1}{2} \frac{dist(a, w_j)^2}{h^2})$$

with

$dim =$ dimension of the random variable X
$a \in \mathbb{R}^{dim}$
$p(a)$: approximation of the density of X at a
R : number of samples for X
$dist(a, w_j)$: Euclidean distance between a and w_j
h : user-defined parameter [20] for changing variance and bias

By integrating over the density function p, we can calculate probabilities for the random variable X. But in our case, we consider discrete systems,

so the random variables (e.g. $Conf_t$) are discrete. If we assume that the random variable only yields integer values for each component (i.e. $X \in \mathbb{Z}^{dim}$), then we can use $P(X = c) = P(dist_\infty(X, c) \leq \frac{1}{2})$ for the approximation, where $dist_\infty(a, b) = \max\{|a_i - b_i| \ : \ i = 1, 2, \ldots, dim\}$ of vectors $a, b \in \mathbb{R}^{dim}$ is the distance induced by the maximum norm on $\mathbb{R}^{dim}$. Since the set $\{a \mid dist_\infty(a, c) \leq \frac{1}{2}\}$ is a hypercube of size 1, the value $P(dist_\infty(X, c) \leq \frac{1}{2})$ can be approximated directly with the density function p, i.e. $P(X = c) \approx p(c)$. This approximation can then be used to get approximations of the entropies $H(\mathrm{Conf}_t)$ and $H(\mathrm{Conf}_t\,|_K)$.

3. Restriction of the set of initial configurations
When we have a system, in which large parts are deterministic, then a restriction of the set of the initial configurations reduces the complexity. Let $\Gamma_0 \subseteq \Gamma$ be a set of initial configurations. Then the time series are received from simulation runs starting in Γ_0. If all automatons are deterministic, two simulation runs with the same initial configuration $c_0 \in \Gamma_0$ would lead to the same time series, so for each initial configuration c_0 at most one simulation run is needed. If some automata are stochastic, the same initial configuration might lead to different time series. The entropy $H(\mathrm{Conf}_t\,|_K)$ (and analogously $H(\mathrm{Conf}_t)$) can then be derived by using the relative frequency $rel_{t,\underline{a}}$ of a value $\underline{a} \in A^K$ at time t as an approximation for the probability $P(\mathrm{Conf}_t\,|_K = \underline{a})$:

$$\begin{aligned} H(\mathrm{Conf}_t\,|_K) &= -\sum_{\underline{a} \in A^K} P(\mathrm{Conf}_t\,|_K = \underline{a}) \log_2 P(\mathrm{Conf}_t\,|_K = \underline{a}) \\ &\approx -\sum_{\underline{a} \in A^K} rel_{t,\underline{a}} \log_2 rel_{t,\underline{a}} \end{aligned}$$

As for the method of classification, this sum can efficiently be calculated, since only the summands with $rel_{t,\underline{a}} \neq 0$ have to be taken into account.

5 Quantitative Measures

In this section we investigate some quantitative measures for global properties, which have been proposed in [1], [2] and [13]. In the following, let $\mathcal{S}$ be a system and (Γ, P_Γ) be an initialization.

Emergence
The level of emergence measures global patterns in the system by considering the dependencies between the valuations of different edges. For a point of time $t \geq 0$ the *level of emergence* at time t is defined by [1]

$$\varepsilon_t(\mathcal{S}, \Gamma) = 1 - \frac{H(\mathrm{Conf}_t\,|_K)}{\sum_{k \in K} H(\mathrm{Conf}_t\,|_{\{k\}})}.$$

The level of emergence is always a value in the interval [0, 1]. If at the current point of time $t \geq 0$ there are large dependencies between the values on the

single edges (which can be seen as patterns), the level of emergence is high: $\varepsilon_t(\mathcal{S}, \Gamma) \approx 1$. If the values of nearly all edges are independent, there will be no pattern, so the level of emergence is low: $\varepsilon_t(\mathcal{S}, \Gamma) \approx 0$. Therefore the map $t \mapsto \varepsilon_t(\mathcal{S}, \Gamma)$ measures the dependencies occurring during the whole run of the system.

To be able to approximate the values $H(\mathrm{Conf}_t|_{\{k\}})$, we use the relative frequencies of the values in the time series for the approximations of $P(\mathrm{Conf}_t|_{\{k\}} = a)$ for $a \in A$. We have the approximation (see Section 4)

$$P(\mathrm{Conf}_t|_{\{k\}} = a) \approx rel_{t,k,a} = \frac{r_{t,k,a}}{R},$$

where R is the number of all time series. For the approximation of $H(\mathrm{Conf}_t|_K)$ we can apply the methods of section 4. This leads to an approximation of the level of emergence. In Section 6 we will calculate the level of emergence for an example system, and in section 7 we will discuss the results.

Target orientation

Before a new system is designed, we have the goal of the system in our mind: The system should fulfill a given purpose. The behavior of each node is defined in such a way, that this goal is reached, so the design of a system needs a target orientation. To measure the target orientation, a valuation map $b : \mathrm{Conf} \to [0, 1]$ for the configurations can be used to describe which configurations are "good": A high value $b(c) \approx 1$ means that the configuration c is a part of our goal which we had in mind during the design of the system. For a point of time $t \geq 0$ the *level of target orientation* [2] of $\mathcal{S}$ at time t is defined by $\mathrm{TO}_t(\mathcal{S}, \Gamma) = E(b(\mathrm{Conf}_t))$, where E is the mean value of the random variable. The level of target orientation measures the valuations $b(c)$ of the configurations during the run of a system.

Assume that we have R time series received from simulation runs. Then we get the approximation $\mathrm{TO}_t(\mathcal{S}, \Gamma) \approx \frac{1}{R} \sum_{j=1}^{R} b(\mathrm{Conf}_{t,j})$, where $\mathrm{Conf}_{t,j}$ is the configuration of the j-th simulation run at time t.

Resilience

To measure the resilience of a system, an automaton $Z_{\theta,v}$ can be used to describe the malfunctioned behavior of a node v. In a computer network, this behavior could be caused by hardware failure or it could be the behavior of an intruder. The system is resilient if despite the malfunctioned nodes the system still runs through many "good" configurations. Let Θ be a set and $p_\Theta : \Theta \to [0, 1]$ be a probability distribution. Let $Z = (Z_{\theta,v})_{\theta\in\Theta, v\in V}$ be a family of stochastic automatons. For $\theta \in \Theta$ let $\mathcal{S}^\theta$ be the system $\mathcal{S}$ after replacing a_v by $Z_{\theta,v}$ for all $v \in V$. Let $(\Gamma^{\mathcal{S}^\theta}, P_{\Gamma^{\mathcal{S}^\theta}})$ be an initialization of $\mathcal{S}^\theta$. Let Conf^θ be the set of the configurations of $\mathcal{S}^\theta$. Let $b = (b_\theta)_{\theta\in\Theta}$ be a family of valuation maps $b_\theta : \mathrm{Conf}^\theta \to [0, 1]$ for the configurations. For a point of time $t \geq 0$ let Conf_t^Θ be the random variable, which applies the random variable Conf_t in the system $\mathcal{S}^\theta$ after choosing $\theta \in \Theta$ randomly according to the probability p_Θ. The *level of resilience* [2] of $\mathcal{S}$ at time t is defined by $\mathrm{Res}_t(\mathcal{S}, \Gamma) = E(b(\mathrm{Conf}_t^\Theta))$, where E is the mean value of the random variable. Therefore the system is resilient

if despite the malfunctioned nodes the system still runs through many "good" configurations.

As for the level of target orientation, we get the approximation $\mathrm{Res}_t(\mathcal{S}, \Gamma) \approx \frac{1}{R} \sum_{j=1}^{R} b(\mathrm{Conf}_{t,j}^{\Theta})$, where $\mathrm{Conf}_{t,j}^{\Theta}$ is the configuration of the j-th simulation run in the changed system at time t.

6 Slot Synchronization in Wireless Networks

In this section we apply the methods of the previous sections to a self-organized slot-synchronization algorithm in wireless networks [21]. The access to the wireless medium is organized in time slots. The distributed algorithm for slot-synchronization is based on the model of pulse-coupled oscillators by Mirollo and Strogatz [22].

In the latter synchronization model, the clock is described by a phase function ϕ which starts at time instant 0 and increases over time until it reaches a threshold value $\phi_{th} = 1$. The node then sends a "firing pulse" to its neighbors for synchronization. Each time a node receives such a pulse from a neighbor, it adjusts its own phase function by adding $\Delta\phi := (\alpha - 1)\phi + \beta$ to ϕ, where $\alpha > 1$ and $\beta > 0$ are constants.

In [21] the pulse-coupled oscillator synchronization model is adapted to wireless systems, where also delays (e.g., transmission delay, decoding delay) are considered.

The discrete micro-level model is described in [1], so we omit the definition of the model here.

Now we compare the results of the quantitative measures calculated by the different approximation methods described in section 4. For each case, we used a complete graph G with 30 nodes with the parameters, which had also been used for the analysis in [21]. Starting from a random initialization, the synchronization usually takes about 500-800 time steps, so for our analysis we use a point in time t, which is greater than 800, such that the nodes have already been synchronized.

Concerning the level of target orientation of $\mathcal{S}$, the good configurations are those, where nearly all nodes work synchronously, so for the valuation b of the configurations we measure the slot distances $dist_c(v, w)$ for $v, w \in V$ in each configuration c. The slot distance is the amount of time elapsed between the beginning of the slot of one node and the beginning of the slot of the other node. The valuation is given by $b(c) = 1 - \frac{\sum_{v,w \in V} dist_c(v,w)}{|V^2| \cdot T/2}$, where T is the length of a slot. The mean value $\mathrm{TO}_t(\mathcal{S}, \Gamma) = E(b(\mathrm{Conf}_t))$ is approximated by the relative frequencies $\mathrm{TO}_t(\mathcal{S}, \Gamma) \approx \frac{1}{R} \sum_{j=1}^{R} b(\mathrm{Conf}_{t,j})$ as described in Section 5. For $R = 300$ simulation runs, the result is $\mathrm{TO}_t(\mathcal{S}, \Gamma) \approx 0.996$, so the system has a very high level of target orientation: After the groups of synchronizations are built, the

[2] See [2].

slot distances are zero for almost every pair of nodes, so $\mathrm{TO}_t(\mathcal{S}, \Gamma) \approx 1$ and therefore the system is target oriented.

Now we consider the level of resilience with respect to an intruder at a node $v_0 \in V$, who wants to disturb the communication. In this case, the parameter set Θ can be used to describe the behavior of the intruder. Here we use Θ as a discrete subset of $\mathbb{R}^+$, where $\theta \in \Theta$ is the duration between two consecutive pulses, that the intruder sends periodically to the neighbors. The system $\mathcal{S}^\theta$ is the system $\mathcal{S}$ after replacing the automaton a_{v_0} by Z_{v_0} and leave all other automatons as they are: $Z_v = a_v$ for $v \neq v_0$. The good configurations are those, where all other nodes are synchronized: $b_\theta(c) = 1 - \frac{\sum_{v,w \in V \setminus \{v_0\}} dist_c(v,w)}{|(V \setminus \{v_0\})^2| \cdot T/2}$. For the complete graph with the parameters, which have already been used above for the target orientation, we calculated the level of resilience with the approximation method of Section 5: $\mathrm{Res}_t(\mathcal{S}, \Gamma) \approx \frac{1}{R} \sum_{j=1}^{R} b(\mathrm{Conf}_{t,j}^{\Theta})$. For $\Theta = \{60, 80\}$ and $R = 300$ the approximated level of resilience is

$$\mathrm{Res}_t(\mathcal{S}, \Gamma) \approx \frac{1}{R} \sum_{j=1}^{R} b(\mathrm{Conf}_{t,j}^{\Theta}) \approx 0.988$$

Therefore the system in this model has a high level of resilience with respect to an intruder, which periodically sends pulses.

Concerning the level of emergence we measure the dependencies between the communications in the system:

$$\varepsilon_t(\mathcal{S}, \Gamma) = 1 - \frac{H(\mathrm{Conf}_t\,|_K)}{\sum_{k \in K} H(\mathrm{Conf}_t\,|_{\{k\}})}.$$

Each pulse is represented by the value $\mathrm{Conf}_t\,|_{\{k\}} = 1$ on the edge k, and the value 0 is used, if no pulse is sent. The values $H(\mathrm{Conf}_t\,|_{\{k\}})$ can be approximated by the relative frequencies of the values in the time series. For the complete graph with 30 nodes we have $|K| = 870$, so for the calculation of $H(\mathrm{Conf}_t\,|_K)$ there exist 2^{870} different edge valuations, which is much too large to be able to calculate the entropy analytically. Therefore we apply the three approximation methods mentioned in Section 4. Let us first consider the classification of nodes. Since all elements $\underline{a} \in [b]$ of an equivalence class get the same probability $P(Conf_t|_K = \underline{a}) \approx \frac{1}{|[b]|} \cdot rel_{t,K_0,[b]}$ for $b = a|_{K_0}$, this methods leads to an increase of the entropy: The relative frequency $rel_{t,K_0,[b]}$ is equally distributed under the elements of $[b]$, while in the time series the random variable $Conf_t|_K$ might hit some elements of $[b]$ more than one time, while other elements do not appear at all as values of $Conf_t|_K$. Therefore this approximation leads to a value $\frac{1}{|[b]|} \cdot rel_{t,K_0,[b]}$, which is higher than the exact value $P(Conf_t|_K = \underline{a})$, where the error depends on the size of the classes: The larger the classes $[b]$, the higher is the approximated value. Table 1 shows the approximations of the level of emergence at time $t = 1000$ in dependency of the parameter $|K_0|$ with $R = 300$ simulation runs. For small sets

Table 1. Approximation of the level of Emergence by classification

$\lvert K_0 \rvert$	870	860	700
$\lvert [b] \rvert$	1	2^{10}	2^{170}
$\varepsilon_t(\mathcal{S}, \Gamma)$	0.988	0.980	0.773

K_0, this method leads to a negative level of emergence, so this approximation method is not useful in this case.

A similar problem appears for the Parzen window approach discussed in section 4: In this case the probabilities are not equally distributed, but the entropy is increased anyhow, because the Gaussian kernel leads to an entropy, which is higher than the exact value of $H(\mathrm{Conf}_t \mid_K)$. But an increase in the parameter h leads to a decrease in the entropy $H(\mathrm{Conf}_t \mid_K)$, so the level of emergence $\varepsilon_t(\mathcal{S}, \Gamma)$ grows with increasing h. Therefore, the right parameter h has to be found to get a good approximation with the Parzen window approach. Table 2 shows the approximations of the level of emergence at time $t = 1000$ in dependency of the parameter h with $R = 300$ simulation runs. Note that a small change in the parameter h has a large impact on the result.

Table 2. Approximation of the level of Emergence by Parzen window

h	0.478	0.479	0.4795	0.48
$\varepsilon_t(\mathcal{S}, \Gamma)$	0.778	0.956	0.981	0.991

Concerning the methods of the restriction of initial configurations, we choose a set $\lvert \Gamma_0 \rvert$ of initial configurations and use the time series of $R = \lvert \Gamma_0 \rvert$ simulation runs. The entropy $H(\mathrm{Conf}_t \mid_K)$ grows with the size $\lvert \Gamma_0 \rvert$, so the level of emergence $\varepsilon_t(\mathcal{S}, \Gamma)$ decreases with increasing $\lvert \Gamma_0 \rvert$. Table 3 shows the results in dependency of the number $\lvert \Gamma_0 \rvert$ of the used initial configurations.

Table 3. Approximation of the level of Emergence by restriction of initial configurations

$\lvert \Gamma_0 \rvert = R$	10	1000	2000
$\varepsilon_t(\mathcal{S}, \Gamma)$	0.99	0.984	0.982

7 Discussion of the Results

Quantitative measures can be defined for the analysis of existing systems and for the design of new systems. Due to the high complexity, it is usually impossible to calculate the exact values of the measures. The main result of this paper is, that we get approximation methods to be able to calculate quantitative measures in self-organizing systems. We investigated different methods for approximations:

- Mean values $E(X)$ of random variables can be approximated by calculating the arithmetic mean value of the samples received from simulation runs.
- Entropies $H(X)$ of random variables are based on probabilities $P(X = w)$ for the values of the random variable, so each method for the approximation of the probabilities leads to an approximation method for the entropy.
- Probabilities $P(X = w)$ of values for random variables can be approximated by relative frequencies. For random variables with a small range of values, this method usually gives good approximations of the probability, while for random variables with a large range of values, other methods are needed, since the relative frequency for all values w might be very low.
  - Classification
    By choosing a classification on the probability space, the problem is reduced to a smaller set of classes, but with a lower accuracy of the result. Since the probabilities are stretched through the elements of the classes, we get a higher entropy. The larger the classes, the more inaccurate is the result for the quantitative measure.
  - Parzen window approach
    An approximation of the density function is derived from the sample values of the random variable by using the Gaussian kernel. The main problem with this approach is to estimate the parameter h for the density function, which is usually not known in advance. A small change in the parameter may lead to a large change in the result of the quantitative measure.
  - Restriction of initial configurations
    By choosing a subset of initial configurations, the accuracy of the approximation of the probabilities with relative frequencies might be increased, especially if large parts of the system are deterministic. Consider for example the approximation of $\mathrm{Conf}_t\,|_K$ for some $t > 0$ by using the relative frequencies of valuations $\underline{a} : K \to A$ for the probability $P(\mathrm{Conf}_t\,|_K = \underline{a})$. If arbitrary initial configurations are used, then it might happen that the random variable $\mathrm{Conf}_t\,|_K$ takes R different values in the R simulation runs, which leads to two different values for the probability: $P(\mathrm{Conf}_t\,|_K = \underline{a}) = \frac{1}{R}$ if $\underline{a}$ is reached in a simulation run at time t and $P(\mathrm{Conf}_t\,|_K = \underline{a}) = 0$ otherwise. By restricting the set of initial configurations, a valuation $\underline{a}$ might be reached in more than one simulation run, so different relative frequencies can be distinguished, which might lead to a better estimation of the entropy than the binary information "$\underline{a}$ is reached" or "$\underline{a}$ is not reached". For systems, where nearly all automatons are nondeterministic, the advantage of this method is very limited, since the problem remains the same: The relative frequencies for all values w might be very low.

These results about the different analysis methods are also confirmed by the case study in Section 6:

- The approximation of the mean value for the target orientation leads to a very high value $\mathrm{TO}_t(\mathcal{S}, \Gamma) \approx 0.996$, which is very close to the exact value,

because the slot distances are zero for almost every pair of nodes after the groups of synchronizations have been built.
- For the approximated value $\mathrm{Res}_t(\mathcal{S}, \Gamma) \approx 0.988$ for the resilience it may be difficult to estimate the exact value, but intuitively it is clear that a single malfunctioned node has only few influence on the whole network with 30 nodes, which form a complete graph, so synchronization between the normal nodes should still be possible, and we can assume that also in this case the exact value for the level of resilience is near 1, so the approximated value has a high accuracy.
- Concerning the emergence, the results are much worse: The values in table 1 indicate, that large classes lead to a decrease of the approximation of the level of emergence, while small classes are useless, because probabilities can not be well approximated by relative frequencies, if some classes are only reached few times in the simulation runs. Table 2 shows that an inaccurate estimation for the parameter h in the Parzen window approach leads to completely different results for the level of emergence. For the method of the restriction of initial configurations, we used $R = |\Gamma_0|$ in Section 6 because all automata are deterministic, so the time series is uniquely determined by the initial configuration.

Table 4 summarizes the advantages and the problems of these methods.

Table 4. Approximation methods for entropy based measures

Method	**Advantage**	**Problems**
Classification	Reduction of state space	Large classes lead to inaccurate results
Parzen window approach	Efficient calculation	Parameter h needs to be known in advance
Restriction of initial configurations	Reduction of relevant states	For accurate results many initial configurations are needed

All approximation methods discussed in this paper are based on time series: By using simulation runs, time series consisting of the configuration sequences can be obtained. Instead of calculating the measures analytically in the model, it is possible to get approximations of the measures directly from the set of time series. Since the model is not needed anymore, this can be generalized to arbitrary time series of configurations: For each set of configuration sequences, the quantitative measure, which were defined analytically in [1], [2] only for the model, can be approximated by considering only the time series. This allows the usage of experimental data from the real world without the need of the model: By measuring the parameters of interest in the real world system, we get some time series, which can be used for the calculation of the quantitative measures.

This fact helps to evaluate existing systems and to compare different systems or different system parameters with respect to self-organizing properties. The results of these evaluations can be used to improve the existing system.

The methods discussed in this paper can be easily applied for other quantitative measures, which have been defined in literature [1] [2] [3]:

- The level of homogeneity [2] is calculated from the entropies of local configurations of each node v, i.e. the information visible at v. Therefore each approximation method for the entropy discussed in Section 4 can be used for an approximation of the level of homogeneity.
- The level of autonomy [1] is calculated from the entropy of the successor configuration at the current point of time. Therefore each approximation method for the entropy discussed in Section 4 can be used for an approximation of the level of autonomy.
- The level of adaptivity [2] is calculated with respect to valuation maps $b_\theta : \mathrm{Conf}^\theta \to [0,1]$, which describe (like for the measure of resilience) which configurations are good and which configurations are bad. As in the case of resilience and target orientation, the mean values of the random variables can be approximated by calculating the arithmetic mean value of the samples received from the simulation run.
- The level of global-state awareness [3] is calculated from the entropy of a random variable describing an equivalence class of initial configurations. Therefore each approximation method for the entropy discussed in Section 4 can be used for an approximation of the level of global-state awareness.

8 Conclusion and Future Work

In this paper we investigated some approximation methods to be able to calculate quantitative measures in a micro-level model of a complex system. The approximation methods are based on time series which can be received from simulation runs of the system. To analyze the practical usability of the concepts, we applied the methods to a slot synchronization algorithm in wireless sensor networks. The main goal of our contribution is to analyze different approximation methods for different quantitative measures with respect to usefulness for practical applications. The case study investigated in Section 6 shows that the measures, which are not based on entropy but on the mean value of random variables, can be well approximated by the methods discussed in this paper. This fact holds especially for the resilience and target orientation. In the case study, a practical application could for example be the optimization of the system parameters α, β in the synchronization algorithm for adjusting the phase function ϕ. The level of target orientation (and analogously for resilience) can be calculated for different values for these parameters to find the optimal values. Another practical application can be found in [3]: The quantitative measure for global state awareness is used to optimize the system parameters of an intrusion detection system.

For measures, which are based on entropies of global properties of the system, the large range of values of the random variables might lead to inaccurate results. These problems are not limited to the investigated example, but appear for many other applications. One possible solution for the method of classification is the choice of "good" classes, where the probability distribution inside each class is nearly uniform, which would lead to more accurate results for the entropy. The characterization of such classes and methods for finding them is left for future work.

References

1. Holzer, R., de Meer, H., Bettstetter, C.: On autonomy and emergence in self-organizing systems. In: Hummel, K.A., Sterbenz, J.P.G. (eds.) IWSOS 2008. LNCS, vol. 5343, pp. 157–169. Springer, Heidelberg (2008)
2. Holzer, R., de Meer, H.: Quantitative Modeling of Self-organizing Properties. In: Spyropoulos, T., Hummel, K.A. (eds.) IWSOS 2009. LNCS, vol. 5918, pp. 149–161. Springer, Heidelberg (2009)
3. Auer, C., Wüchner, P., de Meer, H.: The degree of global-state awareness in self-organizing systems. In: Spyropoulos, T., Hummel, K.A. (eds.) IWSOS 2009. LNCS, vol. 5918, pp. 125–136. Springer, Heidelberg (2009)
4. de Meer, H., Koppen, C.: Characterization of self-organization. In: Steinmetz, R., Wehrle, K. (eds.) Peer-to-Peer Systems and Applications. LNCS, vol. 3485, pp. 227–246. Springer, Heidelberg (2005)
5. Heylighen, F.P.: The science of self-organization and adaptivity. In: Kiel, L.D. (ed.) Knowledge Management, Organizational Intelligence and Learning, and Complexity. The Encyclopedia of Life Support Systems. EOLSS Publishers (2003)
6. Shalizi, C.R.: Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata. PhD thesis, University of Wisconsin-Madison (2001)
7. Nicolis, G., Prigogine, I.: Self-Organization in Non-Equilibrium Systems: From Dissipative Structures to Order Through Fluctuations. Wiley, Chichester (1977)
8. von Foerster, H.: On Self-Organizing Systems and their Environments. In: Self-Organizing Systems, pp. 31–50. Pergamon, Oxford (1960)
9. Ashby, W.R.: Principles of the Self-organizing System. In: Principles of Self-Organization, pp. 255–278. Pergamon, Oxford (1962)
10. Heylighen, F., Joslyn, C.: Cybernetics and second order cybernetics. Encyclopedia of Physical Science & Technology 4, 155–170 (2001)
11. Haken, H.: Synergetics and the Problem of Selforganization. In: Self-organizing Systems: An Interdisciplinary Approach, pp. 9–13. Campus Verlag (1981)
12. Gershenson, C.: Design and Control of Self-organizing Systems. PhD thesis, Vrije Universiteit Brussel, Brussels, Belgium (May 2007)
13. Holzer, R., Wuechner, P., De Meer, H.: Modeling of self-organizing systems: An overview. Electronic Communications of the EASST 27, 1–12 (2010)
14. Di Marzo Serugendo, G., Foukia, N., Hassas, S., Karageorgos, A., Mostfaoui, S.K., Rana, O.F., Ulieru, M., Valckenaers, P., Van Aart, C.: Self-organisation: Paradigms and applications. In: Di Marzo Serugendo, G., Karageorgos, A., Rana, O.F., Zambonelli, F. (eds.) ESOA 2003. LNCS (LNAI), vol. 2977, pp. 1–19. Springer, Heidelberg (2004)
15. Holzer, R., de Meer, H.: On modeling of self-organizing systems. In: Autonomics 2008 (2008)

16. Mnif, M., Mueller-Schloer, C.: The quantitative emergence. In: Proc. of the 2006 IEEE Mountain Workshop on Adaptive and Learning Systems (SMCals 2006), pp. 78–84. IEEE, Los Alamitos (2006)
17. Fisch, D., Jnicke, M., Sick, B., Müller-Schloer, C.: Quantitative emergence a refined approach based on divergence measures. In: Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems, Budapest (2010)
18. Cover, T.M., Thomas, J.A.: Elements of Information Theory, 2nd edn. Wiley, Chichester (2006)
19. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
20. Bishop, C.M.: Novelty detection and neural network validation. In: IEEE Proc. Vision, Image Signal Processing, vol. 141, pp. 217–222 (1994)
21. Tyrrell, A., Auer, G., Bettstetter, C.: Biologically inspired synchronization for wireless networks. In: Dressler, F., Carreras, I. (eds.) Advances in Biologically Inspired Information Systems: Models, Methods, and Tools. SCI, vol. 69, pp. 47–62. Springer, Heidelberg (2007)
22. Mirollo, R., Strogatz, S.: Synchronization of pulse-coupled biological oscillators. SIAM Journal of Applied Mathematics 50, 1645–1662 (1990)

Evolving Self-organizing Cellular Automata Based on Neural Network Genotypes

Wilfried Elmenreich and István Fehérvári

Mobile Systems Group, Lakeside Labs
Institute for Networked and Embedded Systems,
University of Klagenfurt

Abstract. This paper depicts and evaluates an evolutionary design process for generating a complex self-organizing multicellular system based on Cellular Automata (CA). We extend the model of CA with a neural network that controls the cell behavior according to its internal state. The model is used to evolve an Artificial Neural Network controlling the cell behavior in a way a previously defined reference pattern emerges by interaction of the cells. Generating simple regular structures such as flags can be learned relatively easy, but for complex patterns such as for example paintings or photographs the output is only a rough approximation of the overall mean color scheme. The application of a genotypical template for all cells in the automaton greatly reduces the search space for the evolutionary algorithm, which makes the presented morphogenetic approach a promising and innovative method for overcoming the complexity limits of evolutionary design approaches.

Keywords: cellular automata, artificial neural networks, self-organizing systems, evolutionary algorithm.

1 Introduction

The concept of Self-Organizing Systems (SOS), although long known from domains such as physics, chemistry and biology, has recently gained interest to be applied to technical systems. Self-organization can be defined as the emergence of coherent, global behavior out of the local interactions between components. This emergent organization is characterized by intrinsic autonomy, adaptability to environmental changes, and local awareness of the most important global variables. Most importantly, many SOS appear to be robust with respect to a variety of disturbances and intrusions, as the system is to some degree capable of overcoming or self-repairing damages. While many natural, social and technological examples of SOS exhibiting these characteristics are known, and several mechanisms of self-organization have been analyzed in detail, the design of a SOS remains a fundamental challenge [1]. Recent results have shown evolutionary design [2] to be a promising method for designing self-organizing systems [3,4].

However, as the system to be evolved becomes more complex, the evolutionary approach suffers from problems such as disruption of inheritance, premature convergence and failure to find a satisfying solution [5]. Yet, natural evolution has managed to create the very complex design of life. A main difference between natural and artificial evolution is, in many cases, the genotype-phenotype mapping. Natural organisms grow from

C. Bettstetter and C. Gershenson (Eds.): IWSOS 2011, LNCS 6557, pp. 16–25, 2011.

a single cell into a complex system, while in many applications of artificial evolution, there is a one-to-one mapping from genotype to phenotype, leading to poor scalability. Therefore, there is a strong need for introducing generic genotype descriptions that can emerge into arbitrarily complex systems.

In this paper we describe such an approach by the model of a cellular automaton where the state-transition logic of each cell is an instance of the same genotypical controller. The control algorithm is implemented by a small artificial neural network that is evolved to reproduce a given pattern on the cellular automaton. We tested the ability of the resulting pattern formation model for replicating several naturally occurring patterns (such as animal skin patterns) as well as man-made patterns (such as flags and paintings). Results show that the approach works better for regular structures.

The remaining parts of this paper are structured as follows: The following section 2 briefly reviews related work on pattern formation and artificial evolution approaches to evolve pattern formation processes. Section 3 introduces the model, i.e., the Cellular Automata (CA) structure and the properties and interconnections of the Artificial Neural Networks (ANNs). The evolutionary programming method is based on a tool named Frevo, its application for the given problem is described in Section 4. Experiments including evolving several patterns and discussion of the results is elaborated in Section 5. Finally, Section 6 concludes the paper and sketches possible further research and applications based on the presented approach.

2 Related Work

The need for proper genotype descriptions that can be evolved is discussed in the works of Bentley/Kumar [5], Eggenberger [6], and Miller [7]. Bongard and Pfeifer [8] demonstrate a model to evolve the morphology and neural control of an agent.

Amorphous computing [9] refers to systems of many identical simple processors each having limited computational ability and interacting locally. Typically, such systems are of irregular structure in contrast to the regular structure of CAs as they are used in our case study.

Pattern formation in general is studied in developmental biology in terms of cell fate control through a morphogen gradient. A morphogen has been conceptually defined in the 1960s by Lewis Wolpert using the French Flag Model [10] where the colors of the French flag represent the effects of the morphgen on cell differentiation. Herman and Liu [11] have solved the French flag problem through the simulation of linear iterative arrays of cells. Miller [7] has created a simulated differentiated multicellular organism that resembles structure and coloring of the French flag. He uses a feed-forward Boolean circuit implementing a cell program. The cell programs are evolved using a specialized Genetic Programming system. Miller also comments on the difficulty of the problem of evolving a cell program.

Chavoya and Duthen in [12] have used a Genetic Algorithm to evolve Cellular Automata that produce 2D and 3D shapes such as squares, diamonds, triangles, and circles. Furthermore, in [13], they have evolved an Artificial Regulatory Network (ARN) for cell pattern generation, producing the French flag pattern.

Fontana in [14] generated predefined arbitrarily shaped 2D arrays of cells through an evolutionary-developmental technique. He was able to successfully generate complex patterns such as dolphin, hand, horse, foot, frog, and the French flag. Fontana has extended this work in [15] to evolve complex 3D forms.

The combination of neural networks and cellular automata which grow under evolutionary control is present in the China-Brain project [16], an effort to create an artificial brain of interacting small (typically 12-20 neurons) neural networks. This work, having several aspects in common, differs from the work presented in this paper in two aspects: China-Brain is intended as a controller (e.g., to a robot) while in our problem the structure and form of the cells themselves are the output and, second, the interconnections between the modules are designed explicitly by so-called brain architects, while they are an output of the evolutionary process for our model.

3 Cellular Automaton Model

The used model consists of a regular rectangular grid matching exactly the resolution and proportion of the reference image (reference images are very small, typically 50-500 pixels). The colors of the reference image are converted into a scale, where neighboring colors are resembling each other. This color transformation from RGB has been achieved by assembling a binary number by arranging the most significant bits of the channels R,G and B, followed by the second most significant bits of those channels, and so on until the least significant bits, finally yielding a 24-bit color code.

The colors which are present in the reference image are then sorted according to the new scale and assigned to the possible output spectrum of the neural networks. Each color gets an equal proportion of the output space, which is continuous between -1 to 1.

Each cell is controlled by an ANN, which is modeled as a time-discrete, recurrent artificial neural network. Each neuron is connected to every other neuron and itself via several input connectors. Each connection is assigned a weight and each neuron is assigned a bias value.

At each step, each neuron i builds the sum over its bias b_i and its incoming connection weights $w_j i$ multiplied by the current outputs of the neurons $j = 1, 2, ..., n$ feeding the connections. Weights can be any real number, thus have either an excitatory or inhibitory effect. The output of the neuron for step $k + 1$ is calculated by applying an activation function F:

$$o_i(k+1) = F(\sum_{j=0}^{n} w_{ji} o_j(k) + b_i)$$

where F is a simple linear threshold function

$$F(x) = \begin{cases} -1.0 \text{ if } x \leq -1.0 \\ x \quad \text{if } -1.0 < x < 1.0 \\ 1.0 \quad \text{if } x \geq 1.0 \end{cases}$$

In total each ANN consists of 9 input neurons, 5 output neurons and 6 hidden neurons. The 6 hidden neurons have been selected after a short evaluation and turned out to be

an applicable number in order to balance between capability of the ANN and reduction of the search space. One outgoing connection is used to define the cell's color and four pairs of incoming and outgoing neurons connect the ANN with neighboring cells. Furthermore, an ANN can sense the colors of the neighboring cells. The ANN of a cell does not get any direct information about its position in the grid. A cell at a border or in corner, however, would be able to infer about its position. Via the inter-cell connections, information can propagate to the cells in the center.

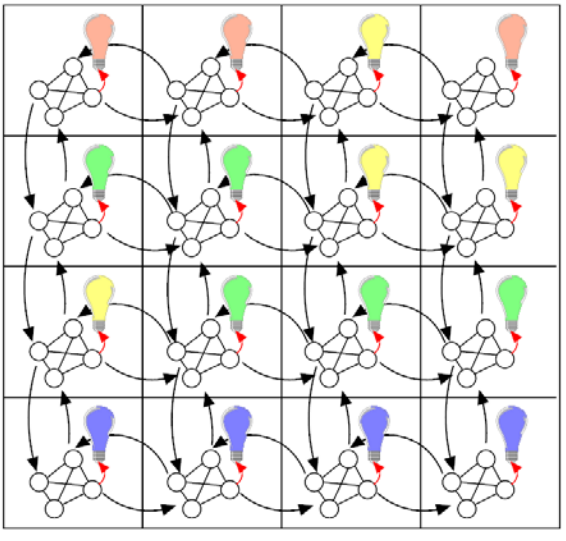


Fig. 1. Interconnections of ANNs in neighboring cells

Figure 1 depicts the interconnections between the ANNs via neighboring cells in the CA model. The light bulb, which is controlled by the ANN in the same cell indicates the current color state of a cell. Each cell gets an instantiation of the same ANN. When the CA is iterated, its ANNs can however acquire a different internal state that can be kept via self-holding loops and that can lead to differentiation in the ANN's behavior.

4 Evolutionary Programming of the ANNs

In order to evolve the weights of the ANN, we used Frevo [3], a Java framework for generating distributed self-organizing systems.[1] Frevo allows to combine different *representations* (e.g., fully-connected ANNs, layered ANNs, Finite State Machines (FSMs)) with an *optimization method* and a *problem* to be solved. Every problem contains a generic interface to one or several instances of the representation, the optimizer basically evolves a control algorithm formulated in the representation that solves the problem.

[1] Frevo and most of its models are available as open source software at `http://www.frevotool.tk/`

The modular concept allowed to reuse the model for the ANN and the optimization algorithm from previous projects, thus reducing our task to formulate and implemented the problem. The used optimization algorithm is depicted in the algorithm below. The selection criteria are based on the rank (according to the candidate's fitness) and, in case of the random selection, also on diversity. Diversity means that candidates which are more different to the already selected pool of candidates have a higher chance to be chosen. Functions for mutation, recombination and the difference between candidates are provided by the specific candidate implementation. In our case, the ANNs applied mutation and recombination on the weights and biases of the artificial neurons. The difference between the candidate is a sum of the squared differences of all weights and biases. The number of neurons and structure of the ANNs was fixed at runtime, since the problem statement has no dynamic aspects and networks with fixed structure have shorter evolution times [17].

Algorithm 1. Evolutionary algorithm used as optimization method

```
create n networks in a population and initialize them with random values

for generations
    for i=0 to n
        evaluate network_p, i and store score
    rank networks according to their score (best first)
    select elitist networks
    select randomly networks (bias for better ranked and diverse networks)
    create mutations of selected networks
    create recombinations of selected networks
    create some networks anew and initialize them with random values
```

The parameters used for the the evolutionary algorithm are listed in Table 1.

Table 1. Parameters used for the evolutionary algorithm

Population size	100
Elite selection	15%
Random selection	10%
Mutated networks	30%
Recombination	40%
New networks	5%
Mutation rate	5%

A problem consists typically of a simulation with a generic interface to the control system. The simulation returns a fitness value which is used by the optimization algorithm for evolving the system. We implemented a CA simulator that is controlled by a representation. The fitness function was implemented as a sum of the squared color index differences between the image after a number of iterations and the reference image. In order to approximate human perception of different images (human vision is focused

around edges rather than areas with the same or similar color), the summands have been weighted by a function giving higher values for pixels having pixels of different color in their neighborhood.

5 Experiments and Results

We found flags to be the simplest kind of patterns to emerge for our proposed set-up. Figure 2 depicts the evolution of a Hungarian flag. The reference image consists of a 6x9 image, where the two top rows are red and the top lower rows are green with white in between. Thus the complexity of this reference is equivalent the French flag which was used as reference in several related work articles as described in Section 2.

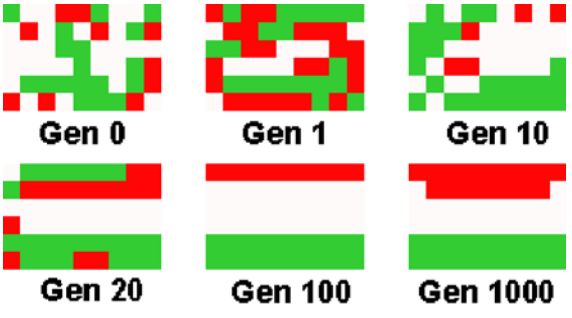


Fig. 2. Evolution of recreating a Hungarian flag

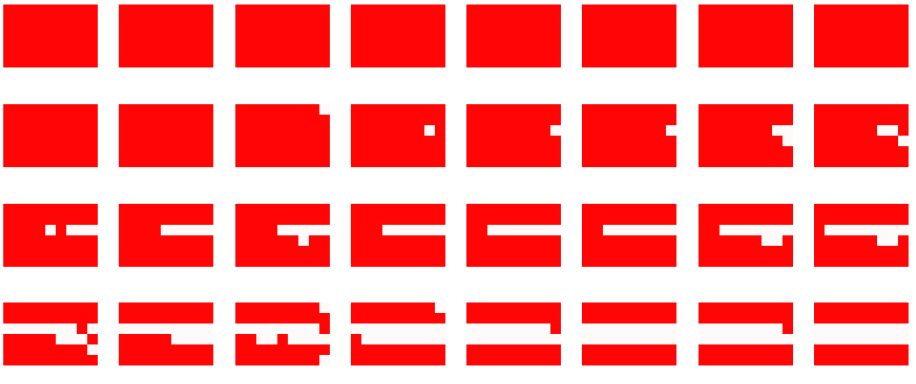


Fig. 3. CA steps for Austrian flag

Note that the proceedings in the quality were highly non-linear over the number of generations, because the evolutionary algorithm gets often stuck in local cost minima after about 100 generations. Thus, improvements past these generations happen only very infrequently.

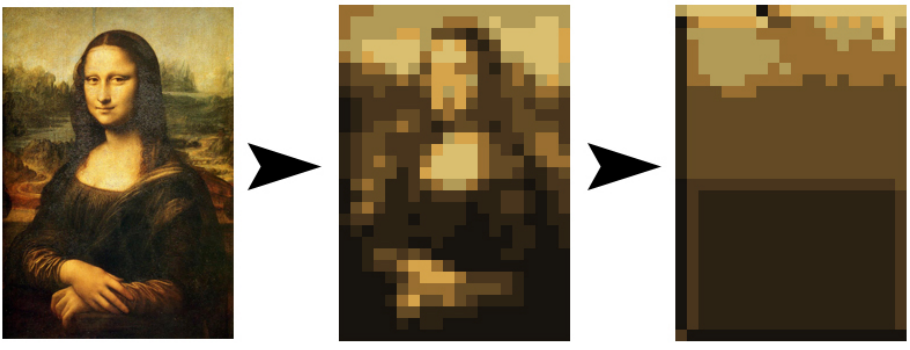


Fig. 4. Attempt to reproduce the work of Leonardo da Vinci

The mechanism to recreate an image over several iterations of the CA can be observed by the example of an Austrian flag. The Austrian flag contains only red and white color and was therefore easier to evolve than the three-colored Hungarian one. We achieved a perfect reproduction after running the evolutionary algorithm for 90 generations. Figure 3 shows how the result unfolds over several CA iterations into the intended image.

The limits of the approach can be observed when going to more complex images. Figure 4 depicts the results of trying to reproduce a small image of the Mona Lisa painting (left image). The middle image shows the downsized reference image. The best achievable result after over 500 generations is depicted in the right image. The overall background color scheme is present, although, unfortunately, Mona is missing. The main reason for this result lies in the increased size of the image – while the flags were evolved on a raster of 6x9, the Mona Lisa image is 20x29. Note that initially only cells at the corner and the borders can detect their position in the image - the inner cells must rely on propagating information. For a larger image, the ratio between border and inner cells is more extreme.

Another question of interest was how well natural patterns can be evolved. Several patterns resembling natural ones can be reproduced by CA executing simple state-transition rules of positive and negative feedback [18].

Interestingly, our evolutionary algorithm did not come up with a feasible solution. This is likely because the fitness function was inappropriate for that task, since it compared the potential solution pixel by pixel to a reference image. Thus, a *similar pattern* is not considered as solution, although a human observer might perceive a similar pattern as being closer to the reference than an image that partially reproduces the original layout of objects in the image. Figure 5 shows that although the created image resembles the reference one on a pixel-by-pixel basis, the quality and type of the reference pattern is not matched.

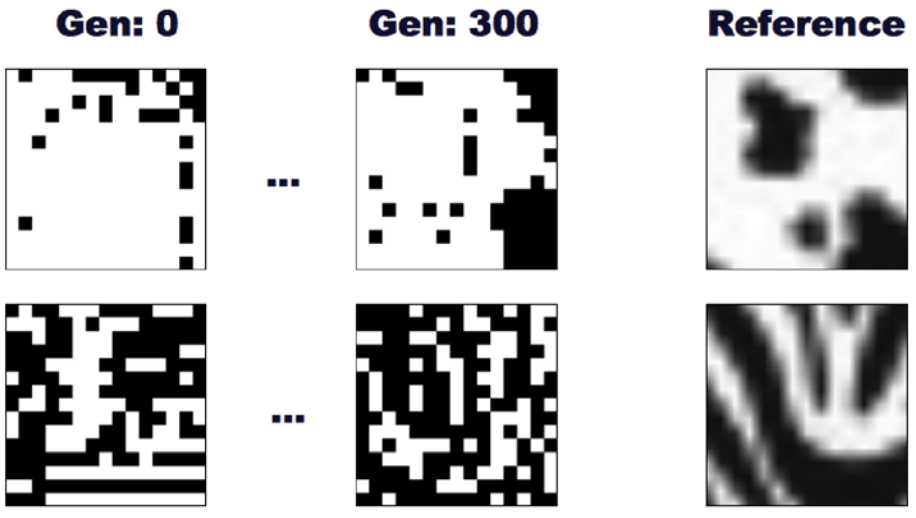


Fig. 5. Evolving a reproduction of animal skin patterns

6 Conclusion and Future Work

We depicted and evaluated a design process that generates a multicellular system out of a genotypical description for a single cell. The mechanisms have been realized via an open source framework for evolutionary design (FREVO). At the beginning of each simulation, all cells had the same state and commenced their operation at the same time - this is comparable with a number of people cooperatively drawing an image in the dark. This differentiates our problem from the ones in the literature, where usually a zygote cell is given, from where the other cells grow. Still, the evolutionary process evolved a solution where also some eminent cell (typically a particular corner cell) serves as a zygote.

The main contribution of this paper is not presenting an algorithm for "drawing images in the dark" but rather presenting a proof-of-concept on integrating ANNs into a CA in order to initiate a morphogenetic process.

Possible applications of this research could be the self-organized pattern formation in swarm robotics. In other words, given a desired pattern, how can robots acquire it? Another application could be smart paint (as indicated in [9]) that would decide on its color based on a morphogenetic process having only a few distinctive sensory inputs, thus not allowing for a zygote approach.

The best results have been achieved when evolving simple structures with large areas of a single color as they are present for example in flags. For more complex images, the current setup causes the evolutionary algorithm to get stuck at a suboptimal stage. There is, however, a large space of possibilities for variations of the model which gives rise to future work. E. g., findings on well-suited or less well-suited model configurations could give insight to the understanding of such phenomena as morphogenesis and camouflage mechanisms in nature.

Future experiments are planned to involve modifications in the internal ANN structure (e.g., investigate on the optimal number of hidden nodes for different reference images) as well as increasing or decreasing the ability of ANNs to communicate with their neighbors. For evolving structures rather than replications of images, we are planning to design the fitness function in a way to have the fitness based on the type of the emerging structure instead of a pixel-by-pixel comparison.

Acknowledgments

This work was supported by the European Regional Development Fund and the Carinthian Economic Promotion Fund (contract KWF 20214|18128|26673) within the Lakeside Labs project DEMESOS and the follow-up project MESON. We would like to thank Marcin Pilat, Miguel Gonzalez, and Rajesh Krishnan for their input on earlier versions of the paper. Furthermore, we would like to thank the anonymous reviewers for their constructive comments.

References

1. Elmenreich, W., Friedrich, G.: How to design self-organizing systems. In: Science beyond Fiction FET 2009, Prague, Czech Republic, pp. 61–62. European Commission: Information Society and Media, Brussels (2009)
2. Bentley, P.J.: Evolutionary Design by Computers. Morgan Kaufmann, San Francisco (1999)
3. Fehérvári, I., Elmenreich, W.: Evolutionary methods in self-organizing system design. In: Proceedings of the 2009 International Conference on Genetic and Evolutionary Methods (2009)
4. Fehervari, I., Elmenreich, W.: Evolving neural network controllers for a team of self-organizing robots. Journal of Robotics 2010, 10 pages (2010)
5. Bentley, P., Kumar, S.: Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 35–43. Morgan Kaufmann, San Francisco (1999)
6. Eggenberger, P.: Evolving morphologies of simulated 3d organisms based on differential gene expression. In: Proceedings of the Fourth European Conf. Artificial Life (ECAL 1997), pp. 205–213 (1997)
7. Miller, J.F.: Evolving a self-repairing, self-regulating, french flag organism. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 129–139. Springer, Heidelberg (2004)
8. Bongard, J.C., Pfeifer, R.: Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny. In: Proceedings of The Genetic and Evolutionary Computation Conference (GECCO 2001), San Francisco, CA, USA, pp. 829–836 (2001)
9. Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight, T.F., Nagpal, R., Rauch, E., Sussman, G.J., Weiss, R.: Amorphous Computing. Communications of the ACM 43(5), 74–82 (2000)
10. Wolpert, L.: Positional information and the spatial pattern of cellular differentiation. Journal of Theoretical Biology 25, 1–47 (1969)
11. Herman, G.T., Liu, W.H.: The daughter of celia, the french flag and the firing squad. In: WSC 1973: Proceedings of the 6th Conference on Winter Simulation, p. 870. ACM, New York (1973)

12. Chavoya, A., Duthen, Y.: Using a genetic algorithm to evolve cellular automata for 2d/3d computational development. In: GECCO 2006: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 231–232. ACM, New York (2006)
13. Chavoya, A., Duthen, Y.: Use of a genetic algorithm to evolve an extended artificial regulatory network for cell pattern generation. In: GECCO 2007: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, pp. 1062–1062. ACM, New York (2007)
14. Fontana, A.: Epigenetic tracking, a method to generate arbitrary shapes by using evolutionary-developmental techniques (May 2008)
15. Fontana, A.: Epigenetic tracking: A possible solution for evo-devo morphogenesis? In: Proceedings of the 1st International Workshop on Morphogenetic Engineering (2009)
16. de Garis, H., Tang, J.Y., Huang, Z., Bai, L., Chen, C., Chen, S., Guo, J., Tan, X., Tian, H., Tian, X., Wu, X., Xiong, Y., Yu, X., Huang, D.: The china-brain project: Building china's artificial brain using an evolved neural net module approach. In: Proceeding of the 2008 Conference on Artificial General Intelligence 2008, pp. 107–121. IOS Press, Amsterdam (2008)
17. Floreano, D., Dürr, P., Mattiussi, C.: Neuroevolution: from architectures to learning. Evolutionary Intelligence 1(1), 47–62 (2008)
18. Bar-Yam, Y.: Dynamics of Complex Systems. Perseus Books, Cambridge (1999)

Self-organized Middle-Out Abstraction

Sebastian von Mammen[1], Jan-Philipp Steghöfer[2],
Jörg Denzinger[1], and Christian Jacob[1,3]

[1] Department of Computer Science, University of Calgary, Canada
[2] Institute of Software & Systems Engineering, Augsburg University, Germany
[3] Department of Biochemistry and Molecular Biology, University of Calgary, Canada
s.vonmammen@ucalgary.ca, steghoefer@informatik.uni-augsburg.de,
denzinge@cpsc.ucalgary.ca, cjacob@ucalgary.ca

Abstract. In this position paper we present a concept to automatically simplify computational processes in large-scale self-organizing multi-agent simulations. The fundamental idea is that groups of agents that exhibit predictable interaction patterns are temporarily subsumed by higher order agents with behaviours of lower computational costs. In this manner, hierarchies of meta-agents automatically abstract large-scale systems involving agents with in-depth behavioural descriptions, rendering the process of upfront simplification obsolete that is usually necessary in numerical approaches. Abstraction hierarchies are broken down again as soon as they become invalid, so that the loss of valuable process information due to simplification is minimized. We describe the algorithm and the representation, we argue for its general applicability and potential power and we underline the challenges that will need to be overcome.

Keywords: Abstraction, multi-agent systems, middle-out modelling, simulation, motif detection, confidence estimation.

1 Introduction

Systems that comprise only few variables can exhibit complex behaviours [18] and due to the multi-facetted interaction networks in natural systems [17] their computation tends to be inefficient. Abstraction is the means to render computations feasible by capturing the essence of what is important on the one hand, and by ignoring those aspects that are seemingly unrelated on the other hand. Thus, the process of abstraction yields either *generalized* or *specialized* models which results in either rather general or very specific results, respectively [9].

However, in models that try to capture the complexity and the nonlinearity of natural systems seemingly unimportant variables can have a major impact under certain circumstances [15]. Hence, contrary to the need for computational efficiency, all available model data should be integrated into one simulation. The solution to this dilemma is a method that automatically adjusts the degree of abstraction of a computational model depending on its expected need of comprehensive calculations. In this position paper, we present the concept of a self-organizing middle-out (SOMO) abstraction system that realizes this

C. Bettstetter and C. Gershenson (Eds.): IWSOS 2011, LNCS 6557, pp. 26–31, 2011.

idea by means of agents that dynamically establish and break down hierarchical relationships.

After a hint at the vast body of work of related ideas (Section 2), we describe the representation and the algorithm that drive our concept (Section 3). In Section 4, we outline the local agent behaviours for pattern-dependent construction and destruction of hierarchies. We summarize our proposed approach and point out open challenges that will need to be addressed in Section 5.

2 Related Work

At the heart of the presented work lies the idea of emergence, i.e. that a group of interacting units, or agents, produces effects that cannot be inferred from the properties and the behaviour of any of the individual agents itself [5]. There are two ways to approach emergent phenomena: either the investigator starts with a simple model of a system's components and their interactions and he wonders what global phenomena might emerge from them, or a global phenomenon has been observed and now the underlying factors need to be unearthed (*inverse problem*). These two cases are similar to bottom-up and top-down design processes: In a bottom-up approach, basic building blocks are combined to result in a higher order design complexity, whereas the top-down approach starts with the desired product of great complexity and follows its stepwise reduction into numerous parts of lesser complexity.

In the context of simulating a 'virtual heart', Sydney Brenner coined the term middle-out [10] as an approach for consistent computation of processes across several levels of detail. So, a middle-out approach works both ways: Locally interacting agents interact in accordance with a given model with the potential to produce emergent phenomena (bottom-up), and the expected phenomena determine the behaviours of the underlying units (top-down). Finding both these approaches in one system, however, is rarely seen. Instead, the idea of building structural and functional complexity from sets of interacting local agents has been attracting some attention. Peter Schuster, for instance, argues that if some agents' interactions nurture themselves, like chemical reactions in hypercycles, such mergers could drive the evolution of complexity [13]. Steen Rasmussen et al. designed a computational model in which, based on interactions in an artificial chemistry, structures form with an increase in structural complexity and with different functionalities, from monomers to polymers to micelles [12]. Alan Dorin and Jon McCormack argued, however, that such phenomena are not surprising given the model's simplicity. In fact, they argue that it takes considerably more effort to determine the novelties brought about by a novel layer in a hierarchy [2].

In general, synthesizing dynamical hierarchies at all scales is a difficult challenge, especially if real emergent, i.e. unforeseeable, features change the behaviours of higher order agents [7]. But even without considering the emergence of global properties or behaviours, the idea of bottom-up learning can prove

useful. Abbas Shirazi et al., for instance, have shown that the effects of interacting agent groups can be learned by means of artificial neural networks and subsumed by according high-order agents to cut computational costs [14].

3 The SOMO Concept

We want to take Shirazi et al.'s approach several steps further. In particular, we envision a multi-agent simulation in which self-organizing individuals form and dissolve abstraction hierarchies (middle-out), incorporating and releasing other agents, based on observed interaction patterns. In general, groups of agents are subsumed by higher-order agents that emulate the groups' interactions. This computational simplification—not all possible interactions are computed—can become invalid at some point in time, as the learned interaction pattern might simply not apply any longer. In this case, the learned hierarchy has to be dissolved, i.e. the adopted agents are released again. Both adoption and release of agents happen recursively, yielding ever-changing, potentially asymmetrical hierarchical structures. Hence, the computational costs over the course of a simulation are only reduced, if the overhead of changing the hierarchy is less costly than the costs for the pruned agent interactions. Therefore, a reduction of computational costs can only be guaranteed, if the learned abstractions have a reasonably long lifetime which is captured in a pattern's *confidence*. Based on the outlined ideas, we term this approach a self-organized middle-out abstraction system, or simply SOMO learner.

3.1 Representation and Algorithm

We choose simple *situation-action pairs* for describing agent behaviours across various levels of scale. Of particular interest in the context of SOMO are *hierarchical operators*, i.e. predicates and actions for coping with hierarchical relationships between agents. In order to establish a hierarchical relationship, an agent might `enter` another agent. Alternatively, it might be `adopted` by another agent. Both actions yield corresponding parent-child relationships between the two agents. Such a parent-child relationship is reverted by `raising` a child in the hierarchy.

We assume that primarily the agents that are root nodes in the hierarchy are considered for execution. Children of a node are only recursively executed, if they are flagged as *active*, which implies the existence of two actions `activate` and `deactivate` to mark an agent accordingly. Deactivated child nodes make sense, for instance, if their parent nodes subsume their behaviours but need to maintain them for state updates and, potentially, their later release and re-activation.

A parent might `cover` its children, rendering them invisible to its environment, or `expose` them. In the latter case, the children are perceivable to other agents at the hierarchical level of their parents and might trigger interactions. The other way round, whenever an active child node is executed, its set of potential interaction partners is limited to its siblings and those agents it is exposed to.

3.2 Interaction Patterns

SOMO agents log the behavioural rules that were activated over a certain period of time in individual *interaction histories*. The entries of the sequential log contain information about the facts that triggered the rule, its effects, and its interaction partners. Similar to [11], we use the interaction histories as databases for finding patterns in the agents' interaction behaviours. Previously unknown patterns, or *motifs*, can be identified in time series relying on various advanced computing techniques such as learning *partial* periodic patterns [4], applying *efficient*, heuristic search [1], *online* motif search [3], and even the identification of patterns of *multiple resolutions* [16]. Motif detection is adapted to interaction histories by assigning symbols, e.g. A or B, to specific log entries and finding patterns in the resulting strings, e.g. $BBABCCBBABDA$. In the given example $BBAB$ is a motif candidate.

The cost of executing agent behaviours can be reduced according to the amount of information provided by the motif. For instance, a motif which relies on a subset of the agent's rules would make it superfluous to consider the remaining rules—the meta-agent would only consider the reduced rule set. Ideally, a motif provides comprehensive information about the interaction partners and the actual interactions, which allows to rewrite the agent rules as efficient sequences of unconditional instructions, with source and target agents readily in place.

SOMO agents can `observe` their own interaction history, as well as those of others, for finding patterns as the basis for building abstraction hierarchies. When `observing` a group of agents, motif detection is applied to their merged interaction history.

4 SOMO Behaviour

By means of the presented representation, the recursive execution of agent hierarchies, and the introduced hierarchical operators, the agents are capable of building and dissolving hierarchies as part of their behaviours. When combined with motif detection in interaction histories, behaviours can be designed to implement the SOMO learner.

4.1 Changing the Hierarchy

In the simplest case, an agent `observes` its own interaction history, `creates` a new agent, `assigns` its own abstracted behaviour, `enters` this new agent and `deactivates` itself. The newly created higher order agent `raises` and `activates` its children and `removes` itself from the simulation, as soon as its confidence has dropped below a certain threshold (see Section 4.2).

More excitingly, as motivated in Section 2, an agent `observes`, `adopts`, `deactivates` a whole group of other agents and `assigns` itself their simplified interaction behaviour. Repeated applications of these learning rules yield continuously growing hierarchies with increasingly simplified behaviours. At the same time, hierarchies are dissolved when no longer appropriate.

4.2 Confidence Estimation

When is it suitable for an `observing` agent to `adopt` another one to build up hierarchies? When will a hierarchy have to be dissolved by `raising` one's children? The key to these questions is *confidence estimation*. There is a large body of work around confidence in statistics [6] and its effective standardization for use in the natural sciences is a vivid research area [8]. The general idea is to estimate the probability that a pattern occurs based on its preceding frequency over a given period of time.

In SOMO, a sufficiently great confidence value leads to abstraction. The confidence value also determines the abstraction's lifespan. Too generous confidence metrics, i.e. too long abstraction lifespans, diminish the accuracy of a simulation. An abstraction agent can be validated by comparing its behaviour to the effects of its children's interaction either at the end of its lifespan or based on heuristics such as the degree of activity in its local environment. In case of miscalculations, the simulation could be reset to a previous simulation state, adjusted and partially recomputed. This additional overhead might make it hard to reach a gain in efficiency. On the other hand, if confidence is assigned too cautiously to motifs, abstraction hierarchies do not get a chance to form in the first place.

5 Discussion and Challenges

We have proposed the algorithmic concept and representation for SOMO, a self-organizing middle-out learner that automatically adjusts process abstractions in multi-agent based simulations. It relies on hierarchical relationships among agents and the fact that agents can change those relationships themselves. Hierarchies of higher order agents are recursively built that subsume and simplify the interaction processes of their children. The actual learning process is performed by motif detection algorithms that work on the agents' locally maintained interaction histories. Each motif is associated with a confidence value that determines the lifespan of a higher level agent.

After the integration of the outlined modules, e.g. motif detection and hierarchical operators, an extensive and systematic investigation into the impact of various system parameters on the accuracy and the efficiency of scientific simulations needs to be started. Following the self-organization paradigm, we aim at an adjustable agent behaviour for building and destroying hierarchies to optimally serve different simulation conditions. For instance, certain subspaces in a simulation might be subjected to fundamental and fast changes, which would result in a loss of efficiency due to the necessary hierarchy management. Other subspaces, however, might be rarely affected by change and abstracting costly agent dependencies could yield a significant gain in performance.

Another challenge is the application of locally learned patterns across the whole simulation space. This could tremendously cut costs for repeatedly observing and learning similar processes. Of course, this idea also emphasizes the more fundamental challenge of how general a motif should be interpreted. Again, this might depend on the volatility of the simulation (sub-)space.

Once the first successful SOMO systems have been explored, it will be important to investigate correlations between the learned abstractions and features and behaviours we find in higher order emergent phenomena. Whether there will be striking similarity or whether these are two completely different aspects of complex systems remains an exciting open question at this point in time.

References

1. Chiu, B., Keogh, E., Lonardi, S.: Probabilistic discovery of time series motifs. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 493–498. ACM, New York (2003)
2. Dorin, A., McCormack, J.: Self-assembling dynamical hierarchies. Artificial Life Eight, 423 (2003)
3. Fuchs, E., Gruber, T., Nitschke, J., Sick, B.: On-line motif detection in time series with swiftmotif. Pattern Recognition 42(11), 3015–3031 (2009)
4. Han, J., Dong, G., Yin, Y.: Efficient mining of partial periodic patterns in time series database. In: Proceedings of the International Conference on Data Engineering, Citeseer, pp. 106–115 (1999)
5. Johnson, S.: Emergence: The Connected Lives of Ants, Brains, Cities, and Software. Scribner, New York (2001)
6. Kiefer, J.: Conditional confidence statements and confidence estimators. Journal of the American Statistical Association 72(360), 789–808 (1977)
7. Lenaerts, T., Chu, D., Watson, R.: Dynamical hierarchies. Artificial Life 11(4), 403–405 (2005)
8. Louis, T.A., Zeger, S.L.: Effective communication of standard errors and confidence intervals. Biostatistics 10(1), 1 (2009)
9. Mitchell, T.: Introduction to Machine Learning. McGraw Hill, Boston (1997)
10. Noble, D.: The music of life. Oxford University Press, Oxford (2006)
11. Ramchurn, S.D., Jennings, N.R., Sierra, C., Godo, L.: Devising a trust model for multi-agent interactions using confidence and reputation. Applied Artificial Intelligence 18(9), 833–852 (2004)
12. Rasmussen, S., Baas, N.A., Mayer, B., Nilsson, M., Olesen, M.W.: Ansatz for dynamical hierarchies. Artificial Life 7(4), 329–353 (2001)
13. Schuster, P.: How does complexity arise in evolution. Complex 2(1), 22–30 (1996)
14. Shirazi, A.S., von Mammen, S., Jacob, C.: Adaptive modularization of the MAPK signaling pathway using the multiagent paradigm. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6239, pp. 401–410. Springer, Heidelberg (2010)
15. Strogatz, S.H.: Nonlinear dynamics and chaos: With applications to physics, biology, chemistry, and engineering. Westview Press, Boulder (2000)
16. Wang, Q., Megalooikonomou, V., Faloutsos, C.: Time series analysis with multiple resolutions. Information Systems 35(1), 56–74 (2010)
17. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393(6684), 440–442 (1998)
18. Wolfram, S.: A new kind of science. Wolfram Media Inc., Champaign (2002)

On the Communication Range in Auction-Based Multi-Agent Target Assignment

Marin Lujak and Stefano Giordani

Dip. Ingegneria dell'Impresa, University of Rome "Tor Vergata", Italy

Abstract. In this paper, we consider a decentralized approach to the multi-agent target assignment problem and explore the deterioration of the quality of assignment solution in respect to the decrease of the quantity of the information exchanged among communicating agents and their communication range when the latter is not sufficient to maintain the connected communication graph. The assignment is achieved through a dynamic iterative auction algorithm in which agents (robots) assign the targets and move towards them in each period. In the case of static targets and connected communication graph, the algorithm results in an optimal assignment solution. The assignment results are compared with two benchmark cases: a centralized one in which all the global information is known and therefore, the optimal assignment can be found, and the greedy one in which each agent moves towards the target with the highest benefit without communication with any other agent.

Keywords: Multi-agent, auction algorithm, target assignment, random geometric graph, network optimization.

1 Introduction

Cooperation of a team of robotic agents is a crucial factor of success of any multi-robot mission and in the target assignment it requires shared information, either via a priori information or communication of information gathered during a mission. In this work, we address a multi-agent (robot) target assignment problem with a team of mobile homogeneous cooperating robotic agents and a set of spatially distributed target locations. Each agent has knowledge of all target positions and only a local information about other agents due to a limited communication range $\rho > 0$; furthermore, it must solve the problem of its target assignment individually through the local interaction with the environment, the communication with connected agents exchanging only its partial, possibly outdated information, and the motion towards the assigned target.

A multi-agent system in which agents have limited communication capabilities achieves an optimal target assignment solution if the global information is up- to-date and at the disposal of all decision makers (agents) at least in a multi-hop fashion; sufficient condition for optimal solution is that the communication graph among agents is connected [20,16]. Recall that a graph is said to be connected, if for every pair of nodes there exists a path between them; otherwise

C. Bettstetter and C. Gershenson (Eds.): IWSOS 2011, LNCS 6557, pp. 32–43, 2011.

it is disconnected and formed of more than one connected component. In general, a connected communication graph is very difficult to obtain and dynamic breaking and (re)-establishment of communication links can be due to many communication imperfections, e.g., link bandwidth, delays, need for encryption, message mis-ordering, as well as range constraints due to physical setup of the network or to the power limitations on communications [4]. The frequency of the communication graph's disconnections depends on the choice of the agents' transmitting range: the larger the range, the less likely it is that the network becomes disconnected. The critical transmitting range (CTR) for connectivity is a minimum common value of the agents' transmitting range that produces a connected communication graph. At CTR, all global data is available to each agent. Under this value, starts the creation of detached connected components and when the value of communication range is zero, the agents communicate only over the target states (free or occupied). If CTR is not high enough and, therefore, the information at the disposal of agents (decision makers) is not up-to-date and complete then we naturally expect to achieve a poor group decision making and lower levels of cooperation and hence assignment performance.

Multi-agent target assignment problem is equivalent to the (linear) assignment problem from the operations research field, where, more generally, the agents compete for targets based on some cost function. Hungarian method or Kuhn-Munkres algorithm is one of the most known polynomial time complexity algorithms that solves the problem in the case of a single decision maker. From the other side, there are also distributed algorithms for the assignment problem, one of which is Bertsekas' auction algorithm [1]. In this algorithm, the targets' prices are regulated by the bids which agents give based on the current prices and potential benefit. The bidding process increases the prices of targets which makes them gradually less attractive to agents. Eventually each agent is assigned to a target and the bidding stops. These as well as many other parallel and distributed algorithms for solving the assignment problem, e.g., [3,5,8,17,20], all assume the complete inter-agent communication graph, the latter with possibly substantial but bounded delays was a subject of [3,8,20].

The main issue we explore in this paper is the fluctuation of the assignment quality in respect to the reduction of the quantity of the information exchanged among agents, the maximum step size towards the targets between two consecutive auction runs, and the contraction of the communication range under the CTR value. We present a modified dynamic version of Bertsekas' auction algorithm [1] in which agents without any a priori assignment information negotiate for and dynamically get assigned to targets based on their local environment observations and the information exchanged only with the agents in their connected component (if any). In this way, agents move from any initial configuration to distinct target positions. The final assignment is optimal if the communication graph described previously is connected. The main idea behind our approach to this problem is to let every agent maximally benefit the presence of the network of communicating agents and mutually exchange the information regarding the targets' assignments so as to increase the assignment quality. The size of the

communication graph among agents is dependent on each agent's communication range and maximum distance of movement between two consecutive communication rounds (maximum step size); the latter is inversely proportional to the frequency of data exchange among communicating agents.

The experimentation of the algorithm is performed in simulated distinct environment cases with the number of agents ranging from 1 to 30. We show through the simulation results that the quality of the assignment solution in the case of the disconnected communication graph is not a monotonic function of agents' communication range and their maximum step size, but is so on the average.

This paper is organized as follows. In Section 2 we present related work. In Section 3, problem formulation and definitions are presented. In Section 4, we recall the minimum weight perfect matching model for the assignment problem and the background of the Bertsekas' distributed auction algorithm. In Section 5, the modified distributed auction algorithm for the case of disconnected communication graph is presented. Section 6 contains simulation results demonstrating the performance of the modified algorithm compared to the greedy and centralized algorithm. We close the work with the conclusions in Section 7.

2 Related Work

Critical transmitting range (CTR) $\rho > 0$ of n randomly employed agents in environment $[0, l]^2$ was a topic of diverse works, e.g., [11,13,14,15]; in sparse ad-hoc networks, it was analyzed in [13,14,15] using the occupancy theory where nodes (agents) are connected with high probability if $\rho = l\sqrt{(c \log l)/n}$ for some constant $c > 0$ [15].

The conditions evaluated in [13] are how many nodes are required and what transmitting ranges must they have in order to establish a mobile wireless ad-hoc network with connectedness in an obstacle free area of limited size; CTR was proven to be $\rho_M = c\sqrt{\ln n/(\pi n)}$ for some constant $c \geq 1$, where n is the number of network nodes (agents) and M is some kind of node mobility [13].

Connected communication graph among agents whether with or without delays in the target assignment was assumed in some recent works, e.g. [8,9,18]. An assignment of a team of vehicles to distinct targets when the assignment cost function is discounted proportionally to the time it takes an agent to complete the task was a topic of [8] where agents communicate over a connected communication network with substantial delays and perform online assignment simultaneously with movements towards their assigned targets. Moore and Passino in [9] modify the Bertsekas' distributed auction algorithm [1] to minimize the loss of benefit that may occur in the target assignment problem when mobile agents cannot remain stationary (e.g., if they are autonomous air vehicles) and may travel away from their assigned task (target) during the assignment interval, which delays and degrades the collective benefit received. [18] addresses dynamical assignment of tasks to multiple agents using distributed multi-destination potential fields that are able to drive every agent to any available destination, and nearest neighbor coordination protocols to ensure that each agent will be assigned to distinct destination.

On the contrary to the assumption of the complete communication graph, [19] deals with the case without communication among the agents; the authors' approach of dynamic assignment problem is to let every agent explore a sequence of targets and eventually be assigned to the first one that is available.

In wireless networks, a frequently used method for inter-agent communication is broadcasting. It is defined as a dissemination mode of data for which the group of receivers is completely open [6]. When a message is disseminated to locations beyond the transmission range, the mobility of the nodes and multi-hopping is used. A simple broadcast scheme assumes a spanning tree of the network nodes which any node can utilize to send a message. Source node broadcasts a copy of the message to each of its neighbors, which broadcast the message to their neighbors, and so on until the leaf nodes (agents) are reached [10]. How well this service performs depends on its availability and the movements of the nodes that participate in the content dissemination [7].

The correlation between the performance of the solution of the distributed multi-agent target assignment problem and the increase of the number of connected components within the communication graph has not been explored so far to the best of our knowledge.

3 Problem Formulation and Definitions

Considering a time horizon made of T time periods, given are set $A = \{1, \ldots, n\}$ of n collaborative mobile robot agents, represented by points in the plane positioned, w.l.o.g., in a square environment $E = [0, l]^2 \subset \mathbf{R}^2$ of side length $l > 0$, with $p_a(t) \in E$ being the position of robot $a \in A$ at time $t = 1, \ldots, T$, and a set $\Theta = \{1, \ldots, n\}$ of n targets (tasks), with $q_\theta \in E$ being the static position of target $\theta \in \Theta$; the latter is seen as mere object without any processing, memory or communication capacities which are all sustained by robotic agents.

Each agent, $a \in A$, is described by the touple

$$a = \{p_a(t),\ \rho,\ d_{max}^{[a]}\}\ , \tag{1}$$

where $\rho \in \mathrm{R}_{>0}$ is a fixed transmitting (communication) range of agent a's wireless transceiver for limited range communication, and $d_{max}^{[a]}$ is its maximum movement distance (maximum step size) in each assignment interval. At any time t, each agent a knows its $p_a(t)$ and the position q_θ of each target $\theta \in \Theta$. Let $c_{a\theta}(t)$ be the (Euclidean) distance between the position of agent a and target θ.

Examples of such an agent-target setup can be found on the production shop floor where targets are production machines, the position of which is globally known to all the agents (mobile robots or automated guided vehicles). This can be also the case in the supply chain management where trucks or any mobile resource of some distribution center has to optimize its group behavior in the distribution of homogeneous goods to fixed locations of clients.

As the agents move through the environment, their sets of connected agents dynamically change due to a limited communication range. In each period t, each agent a is able to communicate to a set of agents $C_a(t) \subseteq A$ (belonging

to the same connected component) reachable in a multi-hop fashion within the communication graph; at any time t, the latter is the random geometric graph (RGG) [2], that is the undirected graph $G(t) = (A, E(t))$ with vertex set A randomly distributed in some subset of R^2, and edge set $E(t)$ with edge $(i, j) \in E(t)$ if and only if

$$\|p_i(t) - p_j(t)\|_2 \leq \rho \,. \tag{2}$$

In this way, two agents which are not within the communication range of each other can communicate over a third agent (communication relay point) in a multi-hop fashion as long as the latter is placed within the communication range of the both. Therefore, agent a together with the set of agents communicating with the same induce a connected subgraph (connected component) of $G(t)$.

We consider the problem of dynamic assignment of set A of robot agents to set Θ of target locations where each agent has to be assigned to at most one target. Total traveled distance by all agents moving towards their targets has to be minimized. We assume that no a priori global assignment information is available and that agents are collaborative and only receive information through their local interaction with the environment and with the connected agents in the communication graph. T is the upper time bound in which all the agents reach their distinct assigned target locations.

When a communication range among the agents participating in the target assignment is lower than CTR, the communication graph is disconnected or even formed of only isolated vertices; in this case the updated information in each period doesn't necessarily arrive to all the agents in the system. The classical Bertsekas' auction algorithm will not guarantee an optimal or even feasible assignment in this context, and certain modifications are necessary. To respond to this open issue, we present a distributed and dynamic version of Bertsekas' Auction algorithm [1]. The latter finds minimum total length routes from agents' initial to distinct final positions (targets) in the multi-agent target assignment problem with a disconnected communication graph. Through this algorithm, we explore the connection between the assignment solution and the sparsity of communication graph varying from a connected communication graph to the case with only isolated vertices.

4 Minimum Weight Perfect Matching Problem

Assuming single decision maker, the problem of finding a feasible assignment of minimum total cost (distance) between agents and targets, can be modeled as the minimum weight perfect matching problem on the weighted bipartite graph $G = (A \bigcup \Theta, E)$ with weight $c_{a\theta}$ on edge $(a, \theta) \in E$ where each target should be assigned to exactly one agent and each agent to exactly one target.

Denoting the above as the primal problem, it is possible to define the dual problem as follows. Let v_θ be a dual variable representing the profit that each agent a will get if it gets matched with the target θ, i.e., v_θ is the value of θ, and u_a the dual variable representing utility of agent a for being matched with

certain target related to agent a. Both variables are real and unrestricted in sign. The objective of the dual problem is

$$\max(\sum_{a} u_a + \sum_{\theta} v_\theta) \ , \tag{3}$$

subject to

$$u_a + v_\theta \leq c_{a\theta} \quad , \forall \ a \ . \tag{4}$$

The dual objective function (3) to be maximized is the sum of the utilities of the agents and the sum of the values of the targets, i.e., the total net profit of the agent-target system. Constraints (4) state that the utility u_a of agent a cannot be greater than the total net cost $(c_{a\theta} - v_\theta)$ of allocation (matching) to target θ, and each agent a would logically want to be assigned to the target θ_a with minimal value of the total net cost,

$$c_{a\theta_a} - v_{\theta_a} = \min_{\theta}(c_{a\theta} - v_\theta) \ , \ \forall a \ . \tag{5}$$

Agent a is assumed happy if this condition holds and an assignment and a set of prices are at equilibrium when all agents are assumed happy. Equilibrium assignment offers minimal total cost and thus solves the assignment problem while the corresponding set of prices solves an associated dual optimization problem. On the base of the duality in linear programming, $\sum_\theta v_\theta + \sum_a u_a \leq \sum_{a\theta} c_{a\theta} \cdot x_{a\theta}$, where $x_{a\theta}$ represents the assignment variable equal to 1 if agent a is assigned to target θ, and 0 otherwise. Therefore, the total net profit of the system cannot be greater than the total assignment cost that agents have to pay for moving to targets, and only at the optimum, those two are equal [1].

Agent a is almost happy with an assignment to target θ_a and a set of target values if the total net cost of being assigned to target θ_a is within ϵ (a positive scalar) of being minimal, that is,

$$c_{a\theta_a} - v_{\theta_a} \leq \min_{\theta}(c_{a\theta} - v_\theta) + \epsilon \ , \forall a \ . \tag{6}$$

The condition (6) is known as ϵ-complementary slackness. An assignment is almost at equilibrium when all agents are almost happy.

In the forward auction algorithm, which we treat in this paper, v_θ can only increase in each iteration and period. For a detailed description of the Bertsekas' forward auction algorithm we invite the reader to refer to the work [1].

5 Dynamic Online Auction with Mobility

In the following, we propose a partially asynchronous algorithm to solve the distributed target assignment problem. We integrate n copies of the modified auction algorithm which is run online by every agent as it moves toward its assigned target in each period. Each agent a keeps in its memory the value $v_{\theta a}$, that is, its most recent knowledge about the actual value of target θ, for each $\theta \in$

Θ, and the set S_a of its most recent knowledge about all the agents' assignments. Both $v_{\theta a}$ and S_a do not have to coincide with the actual target value and agents' assignments, respectively; they may also differ from one agent to another due to the dynamics of their previous communication and local interaction with the environment.

- Initially, i.e., at time $t = 1$ and auction iteration $h = 0$, for each robot agent $a \in A$, set $S_a(t = 1, h = 0)$ of assignments is assumed empty and all target values $v_{\theta,a}(1, 0)$ are set to zero.

At each time $t \in [1, \ldots, T]$, a new round of iterative auction is performed, starting from the assignments and target values locally available to the agents from the end of the round before. In more detail, during iteration h of round t:

- each agent a broadcasts and receives target values $v_{\theta,a}(t, h-1)$ and assignments $S_a(t, h-1)$ to/from all the agents within the connected component the agent a belongs to within the communication graph.
- Each agent updates its local list of the assignments $S_a(t, h)$ and target values $v_{\theta,a}(t, h)$ by adopting the largest value $v_\theta(t, h-1)$ among the set of agents $C_a(t)$ within the connected component for each target of interest $\theta \in \Theta$ and the assignment resulting from this value. However, if the target θ_a is assigned to more than one agent, its assignment is canceled making it unassigned and eligible for bidding in the ongoing round unless if it has already come to its target position q_{θ_a}; in that case, the agent on the target position remains assigned to the target while other agents within the connected component update the value of the target θ_a and cancel the assignment to the same.
- If agent a is unassigned, it finds its best target, calculates the bid value and bids for that target using the following bidding and assignment procedure.

5.1 Bidding

To submit a bid, each agent a unassigned in its partial assignment $S_a(t, h)$:

- finds target θ_a which offers the best possible value $\theta_a = \arg\min_{\theta \in \Theta}\{c_{a\theta}(t) - v_{\theta a}(t, h)\}$, and calculates bid for target θ_a as follows: $b_{a\theta_a}(t, h) = v_{\theta_a a}(t, h) + u_a(t, h) - w_a(t, h) + \epsilon = c_{a\theta_a}(t) - w_a(t, h) + \epsilon$, where $u_a(t, h) = \min_{\theta \in \Theta}\{c_{a\theta}(t) - v_{\theta a}(t, h)\}$ and $w_a(t, h) = \min_{k \neq \theta_a \in \Theta}\{c_{ak}(t) - v_{ka}(t, h)\}$ is the second best utility that is, the best value over targets other than θ_a.
- raises the value of its preferred target by the bidding increment γ_a so that it is indifferent between θ_a and the second best target, that is, it sets $v_{\theta a}(t, h)$ to $v_{\theta a}(t, h) + \gamma_a(t, h)$, where $\gamma_a(t, h) = v_{\theta a}(t, h) - w_a(t, h) + \epsilon$.

The bidding phase is over when all the unassigned agents calculate their bid.

5.2 Assignment

Let $P(\theta_a)(t, h) \subseteq C_a(t)$ be the set of agents with bids pending for target θ_a. Only one agent $a_{coord} \in P(\theta_a)(t, h)$ is responsible for the assignment of target θ_a and

if there is more than one agent in $P(\theta_a)(t,h)$, the first one in the lexicographic ordering coordinates the auction for that target.

Each agent $a \neq a_{coord} \in P(\theta_a)(t,h)$, broadcasts its bid $b_{a\theta_a}(t,h)$. Agent a_{coord} receives the bids $b_{k\theta_a}(t,h)$ of all other agents $k \in P(\theta_a)(t,h)$, $k \neq a_{coord}$, regarding θ_a. Following steps are performed to resolve the assignment:

- Agent a_{coord} selects agent $a_{\theta_a} = \arg\max_{a \in P(\theta_a)(t,h)} b_{a\theta_a}$ with the highest bid $b_{a\theta_{a\max}} = \max_{a \in P(\theta)(t,h)} b_{a\theta_a}$.
- If $b_{a\theta_{a\max}} \geq v_{\theta_a a}(t,h) + \epsilon$ then $v_{\theta_a a}(t,h) := b_{a\theta_{a\max}}$, the updated assignment information is broadcasted to all the agents $k \neq a_{coord} \in C_a$ which update their sets of assignments S_a by replacing the current agent assigned to it (if any), with agent a_{θ_a}.
- If $b_{a\theta_{a\max}} < v_{\theta_a a}(t,h) + \epsilon$ then all bids for target θ_a are cleared, no reassignment or target value change is made.

If there are any unassigned agents left within C_a, the assignment algorithm starts again from the bidding phase within iteration $h+1$. This process terminates when each agent $a \in A$ has a target assignment.

5.3 Agent Movement

When all agents are assigned to their respective targets in the final auction iteration, say $h_{fin}(t)$, at round t of the iterative auction, the agent movement phase takes place:

- if agent $a \in A$ is not at its target position p_{θ_a}, it moves one step toward θ_a, covering at most a maximum distance $d_{max}^{[a]}$.
- Once when agent a comes to the position of its assigned and available target, it sets its target value $v_{\theta_a a}(t, h_{fin}(t))$ to $-\inf$, and broadcasts the latter in C_a; this will disable assigning the rest of agents to the same target.

If the initial targets' values $v_\theta(1,0)$ are identically zero, the ϵ-CS condition of (6) will be satisfied because the value of any assigned target is strictly increasing during the auction process and must be at least ϵ. To assure the convergence of the algorithm and to exclude the possibility of multiple exchanges of agents on the target positions, and therefore of infinite loops, we model the target value in the way that it decreases infinitely when an agent arrives on its assigned available target position so that once arrived, an agent remains on the same. Since there is a limited number of bids for any target while there still exist targets that have not yet received any bids, the dynamic auction algorithm will finish when all targets have been given at least one bid and the last agent comes to its target.

The presented algorithm works also when the number of agents differs from the number of targets. In fact, if the number of targets is not less than the number of agents, the assignment process terminates when each agent reaches its assigned target; otherwise, the assignment process terminates when all the unassigned agents realize that all the targets are occupied.

6 Simulation Setup and Results

We simulate a Multi-Agent (robot) System with the forward auction algorithm implemented in MatLab. The dynamic modified auction algorithm was experimented with 4 different kinds of information exchange among agents within each $C_a(t)$, $\forall a \in A$, $t \in \{1, \ldots, T\}$, and $h \in \{0, \ldots, h_{fin}(t)\}$:

- exchange of all assignment data in S_a and target values $v_{\theta,a}(t,h)$ for targets $\theta \in \Theta$ among all agents $a \in C_a$;
- partial exchange of target values $v_{\theta_a,a}(t,h)$ and assignments in S_a regarding only the targets assigned to connected agents $a \in C_a$;
- exchange only of the target value $v_{\theta_a,a}(t,h)$ among the agents assigned to the same target θ_a;
- no information exchange at all. The result is a greedy algorithm where each agent behaves in a selfish and greedy way.

W.l.o.g., and for simplicity, we model the agents as points in plane which move on a straight line towards their assigned targets. Experiments were performed for up to 30 agents in $[0, 100]^2 \subset \mathbf{R^2}$ where the initial robot agent and target positions were generated uniformly randomly. The value of communication range ρ is set from 0 to 50 since a critical communication range for 30 agents calculated by the CTR formula in [13] is 24; furthermore, maximum step size d_{max} varies from 1 to 70 since above the latter value, the number of exchanged messages, crossed distance, and the number of algorithm runs remain unchanged. For each number of robots n varying from 1 to 30, we considered 10 different instances.

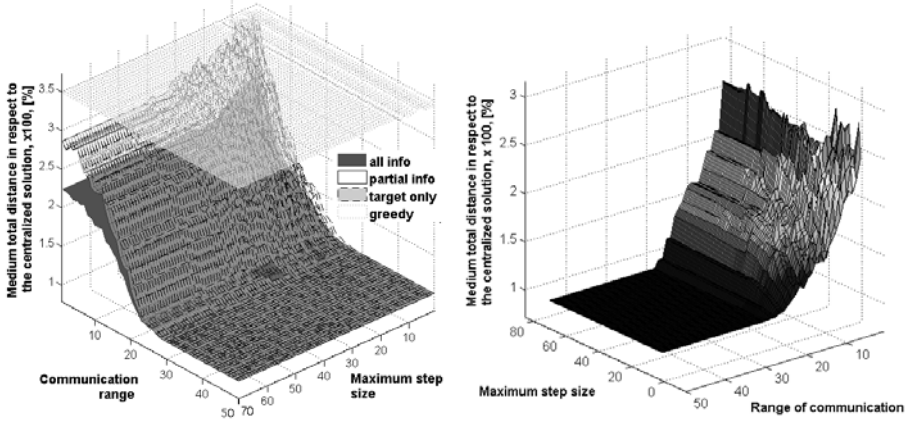


Fig. 1. Left: medium total distance (relative to optimal solution) crossed by 30 agents in $[0, 100]^2$ environment in respect to communication range ρ and maximum step size $d_{max}^{[a]}$; right: assignment dynamics of one representative instance with 30 agents and all information exchange in respect to communication range ρ and max step size $d_{max}^{[a]}$ in the range $[1, 70]$

The average assignment solution (over 10 instances) for the problem with 30 agents is presented in Figure 1, left. The latter represents the variation of the assignment solution in respect to the optimal case with varying maximum step size $d_{max}^{[a]}$ and the communication range ρ.

The presented auction algorithm with included mobility is stable, i.e., it always produces a feasible solution. The experiments show that the average total crossed distance for all the 3 cases with information exchange give the same optimal solution as Bertsekas' algorithm in the case of connected communication graph among all the agents; furthermore, in all of these cases, the solution degrades gracefully with the decrease of the communication range ρ and the increase of the maximum step size $d_{max}^{[a]}$. In the case when the communication range is lower than CTR, the previously mentioned cases give very unexpected results. The number of messages exchanged among the robots in all of the communication models is limited by $O(n^3)$ (Figure 2, right).

In the model with all the assignment information exchange, the assignment solution is, as expected, the closest to the optimal solution, and in the worst case, when the communication range is close to zero and the maximum step size is higher than 60, the average total distance crossed by agents comes up to 260% of the optimal solution.

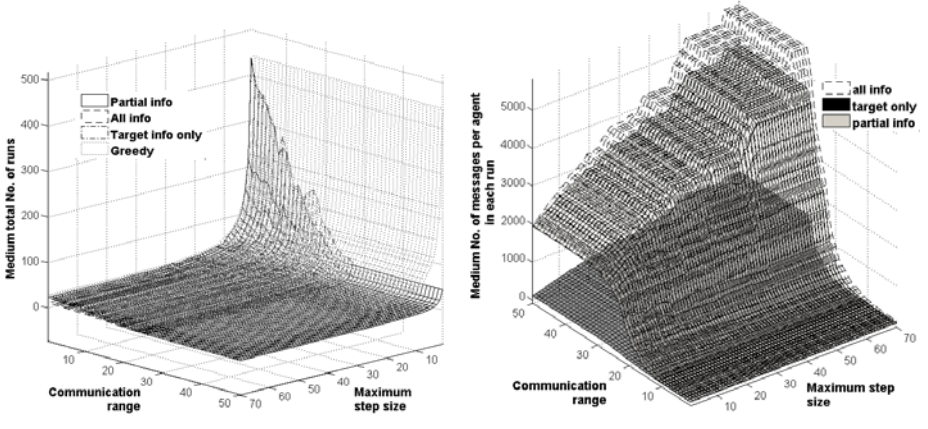


Fig. 2. Left: Medium total number of runs of the algorithms with exchange of all of the information (dashed) vs. partial exchange of information (solid), target only (dash-dot), and greedy (dots); right: medium number of messages per agent in each run for the algorithm with all information exchange (dashed) vs. the one with partial (solid), and target only (black) information exchange

The highest number of exchanged messages per agent and per run, (5733), however, is achieved by the model with all the information exchange. This large amount of information that must be passed over the agents' communication network is a potential problem if the latter is subject to imperfections such as delays, noise, and changing topology.

Unexpectedly, the model with partial information exchange and the model with exchange of information regarding only own target show very close crossed distance results with the worst case approaching 339% and 350% respectively, while in the terms of the medium number of exchanged messages per agent in each run, the model with partial information exchange is up to 4 times more demanding than the target only model.

It is important to notice on the Figure 1, right, that the increase of the communication range when the communication graph is not connected, doesn't necessarily give a better assignment solution; this was the case in all the simulated problem instances.

Considering the communication imperfections, it might be useful to forgo optimality of the assignment for the stability by settling for suboptimal solution by only exchanging the own target information with the agents assigned to the same. The reason not to use a greedy approach is that all of the presented models show, however, better performance than the greedy algorithm which is an upper bound on the average total distance crossed with the value of 351% of the optimal solution, and on the average number of runs, with the worst case of 443 (Figures 1, left, and 2, left). Even if the decentralized method with only target information exchange is cumulatively heavier at the level of exchanged messages (Figure 2, right), the running time of each robot (in the order of $O(n^2)$) results minor in any case than the time required by the greedy algorithm (Figure 2, left).

7 Conclusions

A decentralized implementation of the Bertsekas' auction algorithm has been presented to solve a Multi-Agent (robot) Target Assignment problem for the case when there is no centralized controller or a shared memory and a connected communication graph among agents is not available due to low inter-agent communication range. The agents must operate autonomously and the decentralized implementation is possible on the basis of an exchange of messages. We have assumed homogeneous agents, but the case where agents can be assigned to a specific subset of targets can be handled with a proper elimination of targets within their local assignments. Furthermore, dynamic appearance of targets is supported by the proposed algorithm as long as the positions of targets are known to each agent in each period. The cumulative execution time of the decentralized algorithm and the total number of messages exchanged are in the order of $O(n^2)$, and $O(n^3)$ respectively, resulting in an average computational payload $O(n^2)$ for each agent. A decentralized implementation is intrinsically robust: how robot failures can be handled will be extended in future research.

References

1. Bertsekas, D.P.: Auction algorithms for network flow problems: A tutorial introduction. Comput. Opt. and Applications 1(1), 7–66 (1992)
2. Díaz, J., Mitsche, D., Pérez-Giménez, X.: On the connectivity of dynamic random geometric graphs. In: Proc. of 19th ACM-SIAM Symposium on Discrete Algorithms, pp. 601–610 (2008)

3. Finke, J., Passino, K.M., Sparks, A.: Cooperative control via task load balancing for networked uninhabited autonomous vehicles. In: Proc. of 42nd IEEE Conference on Decision and Control, vol. 1, pp. 31–36 (2003)
4. Gil, A.E., Passino, K.M., Ganapathy, S., Sparks, A.: Cooperative scheduling of tasks for networked uninhabited autonomous vehicles. In: Proc. of 42nd IEEE Conference on Decision and Control, pp. 522–527 (2003)
5. Giordani, S., Lujak, M., Martinelli, F.: A Distributed Algorithm for the Multi-Robot Task Allocation Problem. In: García-Pedrajas, N., Herrera, F., Fyfe, C., Benítez, J.M., Ali, M. (eds.) IEA/AIE 2010. LNCS, vol. 6096, pp. 721–730. Springer, Heidelberg (2010)
6. Keshavarz, H.A., Ribeiro, V., Riedi, R.: Broadcast capacity in multihop wireless networks. In: Proc. of 12th Int. Conf. on Mob. Comp. & Netw., pp. 239–250 (2006)
7. Korkmaz, G., Ekici, E., Özgüner, F., Özgüner, Ü.: Urban multi-hop broadcast protocol for inter-vehicle communication systems. In: Proc. of 1st ACM Int. Workshop on Vehicular Ad-hoc Networks, pp. 76–85 (2004)
8. Moore, B.J., Passino, K.M.: Coping with information delays in the assignment of mobile agents to stationary tasks. In: IEEE Conference on Decision and Control, Institute of Electrical and Electronics Engineers, UK, pp. 3339–3344 (2004)
9. Moore, B.J., Passino, K.M.: Distributed task assignment for mobile agents. IEEE Transactions on Automatic Control 52(4), 749–753 (2007)
10. Murphy, A.L., Picco, G.P.: Reliable communication for highly mobile agents. Aut. Agents and Multi-Ag. Sys. 5(1), 81–100 (2002)
11. Penrose, M.: Random geometric graphs. Oxford University Press, USA (2003)
12. Ren, W., Beard, R.W., McLain, T.W.: Coordination Variables and Consensus Building in Multiple Vehicle Systems. In: Coop. Control: a Post-Workshop: Block Island Workshop on Coop. Control, pp. 171–188 (2004)
13. Santi, P.: The critical transmitting range for connectivity in mobile ad hoc networks. IEEE Trans. on Mob. Comp. 4(3), 310–317 (2005)
14. Santi, P., Blough, D.M.: An evaluation of connectivity in mobile wireless ad hoc networks. In: Proc. of Int. Conf. on Depend. Sys. and Netw., pp. 89–98 (2002)
15. Santi, P., Blough, D.M., Vainstein, F.: A probabilistic analysis for the range assignment problem in ad-hoc networks. In: Proc. of ACM Mobihoc, pp. 212–220 (2001)
16. Savkin, A.V.: The problem of coordination and consensus achievement in groups of autonomous mobile robots with limited communication. Nonlinear Analysis 65(5), 1094–1102 (2006)
17. Smith, S., Bullo, F.: A Geometric Assignment Problem for Robotic Networks. In: Modeling, Estimation and Control, pp. 271–284. Springer, Heidelberg (2008)
18. Zavlanos, M.M., Pappas, G.J.: Dynamic assignment in distributed motion planning with local coordin. IEEE Trans. on Rob. 24(1), 232–242 (2008)
19. Zavlanos, M.M., Pappas, G.J.: Sensor-based dynamic assignment in distributed motion planning. In: IEEE Int. Conf. on Rob. and Autom., pp. 3333–3338 (2007)
20. Zavlanos, M.M., Spesivtsev, L., Pappas, G.J.: A distributed auction algorithm for the assignment problem. In: Proc. of 47th IEEE Conf. on Decision and Control, pp. 1212–1217 (2008)

An Adaptive Control Technique for a Connection Weight of Agents in a Self-repairing Network

Masahiro Tokumitsu[1] and Yoshiteru Ishida[2]

[1] Department of Electrical and Information Engineering,
Toyohashi University of Technology,
Hirbarigaoka 1-1, Tempaku, Toyohashi, Aichi, 441-8580, Japan
tokumitsu@sys.cs.tut.ac.jp
[2] Department of Computer Science and Engineering,
Toyohashi University of Technology,
Hirbarigaoka 1-1, Tempaku, Toyohashi, Aichi, 441-8580, Japan
ishida@cs.tut.ac.jp
http://www.sys.cs.tut.ac.jp/

Abstract. Cooperation among agents is a crucial problem in autonomous distributed systems composed of selfish agents pursuing their own profits. An earlier study of a self-repairing network revealed that a systemic payoff made the selfish agents cooperate with other agents and was similar to kin selection. We study the relationship between the systemic payoff and kin selection more deeply. This paper considers the systemic payoff that involves a connection weight representing strength of relationship among the agents. We found that the performance of the self-repairing network changes by varying the connection weight. The connection weight appropriate to the environments would elicit the good performance of the self-repairing network. This paper proposes an adaptive control technique for the connection weight among the agents in the systemic payoff. The technique changes the connection weight dynamically. In simulations, the proposed technique showed the good performance in the dynamic environments.

Keywords: autonomous distributed systems, kin selection, game theory, selfish agents, self-repairing network.

1 Introduction

Autonomous distributed systems are composed of selfish agents pursuing their own profits. In the autonomous distributed systems, the selfish agents need to cooperate with other agents because the collective selfish acts of the agents would lead the systems to the absorbed states. A study on selfish routing reported that if the agents route their traffic selfishly then the network would show the bad performance [1,2]. Cooperation is a crucial issue in the autonomous distributed systems. Many earlier studies have proposed the cooperation mechanism to make the agents cooperate with each other.

C. Bettstetter and C. Gershenson (Eds.): IWSOS 2011, LNCS 6557, pp. 44–55, 2011.

We can observe the autonomous distributed systems in various fields. For example, biological systems [3,4]. Social insects such as honeybees show altruistic behaviors to their colleagues in its colonies. The offsprings of queen bees take care of their brothers and sisters. The reason of these phenomena is explained by *kin selection* [3,4]. Kin selection is a model that explains why related individuals cooperate with each other. An inclusive fitness of kin selection involves not only an individual's own payoff but also other individual's payoff. Those systems tell us that cooperation is the important factor of the autonomous distributed systems.

Cooperation is an important problem in the self-repairing network [5,6]. The self-repairing network is the theoretical model that the agents repair other agents mutually [7]. It is composed of the selfish agents. In the self-repairing network, to bring out cooperation among the agents has been studied using spatial strategies and the payoff mechanism.

Earlier studies [5,6] revealed that the systemic payoff is capable of making the agents cooperate with other agents in the self-repairing network. Moreover those studies reported that the systemic payoff was similar to kin-selection. The systemic payoff sums up not only its own payoff but also the neighbor's payoff connected. Those studies concluded that the agents with the systemic payoff improved the network performance. However, the relationship between the systemic payoff and kin selection has not been studied deeply. Furthermore, the study on how much the agents include the neighbor's payoff to its own profit has not been investigated well.

This paper has three purposes. The first aim is to study the relationship between the systemic payoff and kin selection. This paper constructs a mathematical model of the self-repairing network that incorporates the systemic payoff. We consider the systemic payoff involving the connection weight of the agents. Then, we study the relationship between the systemic payoff and kin selection.

The second aim is to investigate the effect of the systemic payoff for the performance of the self-repairing network. We examine the self-repairing network with the modified systemic payoff by numerical simulations. We carry out the numerical simulations of the model in the static environments.

The third aim is to propose an adaptive control technique for a connection weight of a systemic payoff. The agents need to adjust its connection weight in order to adapt to the environments. A related study [8] introduced altruism to the agents in a traffic routing system. Our technique differs from that the connection weight changes dynamically. Furthermore, the technique is a decentralized approach that each agent controls its connection weight by using the averaged resources. We evaluate effectiveness of our proposed technique in numerical and agent simulations.

2 Self-repairing Network Model

We model the self-repairing network by using game theory. The game theory is used to analyze the decision making of the agents interacting with each other. In

the self-repairing network, the agents make the decisions *repair* or *not repair*. The abnormal agents will spread among the network if no agent repairs themselves. However, the agents are reluctant to consume their resources for repairing because of they are eager to assign the resources to their own tasks. We can model this situation by Prisoner's Dilemma [9]. The actions of the agents *repair* or *not repair* correspond to the actions *cooperation* or *defection* in Prisoner's Dilemma.

We model the self-repairing network by Spatial Prisoner's Dilemma [9] because the decision making of the agent is affected by spatial arrangement of the agents. The Spatial Prisoner's Dilemma is a spatial extension model of the Prisoner's Dilemma. We consider the model involving the spatial interactions of the agents by the Spatial Prisoner's Dilemma.

We consider the simple model of the self-repairing network for describing the dynamics. We assume that the network is large and infinite agents exists. The agents interact with each other randomly. The agents have states: normal or abnormal. Each agent determines the next action: *repair* or *not repair*. We denote the frequency of normal agents (abnormal agents) by ρ_N (ρ_A).

The agents determine their actions based on their strategies. The agents choose either All-C or All-D strategies. The All-C strategy always repairs other agents, while the All-D does not repair. We denote the frequency of the All-C (All-D) agents by ρ_C (ρ_D).

The repairing is done between the two agents. The two agents are randomly chosen from the network, then they repair each other based on their decisions. The repair success rate is different by the state of the agent. We denote the repair success rate of the normal and abnormal agents by α and β respectively. We assume to simplify the model that the repair by the normal agents is always successful ($\alpha = 1$). The repaired agent becomes normal if the repairing is successful, otherwise the repaired agent becomes abnormal. We assume that the normal agents become abnormal by spontaneous failure. We denote the failure rate by λ.

The agents have their own resources consumed by repairing. The normal agents (abnormal agents) have the maximum resources b_N (b_A). If the normal agents (abnormal agents) do the repairing, then the agents consume their resources only c_N (c_A). The agents deal with the remained resources as the available resources. The resources of the agents are filled in the beginning of the step.

3 Dynamics of Self-repairing Network

3.1 Replicator Dynamics of Self-repairing Network

We describe the dynamics of the self-repairing network by replicator equations [10] and master equations. The replicator equations are a mathematical model in evolutionary game theory and can model among interactions between the agents based on the strategies and the payoff matrix. The states of the agents affect the success of the repair. We model the repairing dynamics of the agents by master equations.

The expected payoffs of the strategies can be calculated by summing up the payoffs of the agents in each state. The expected payoffs can be expressed as follows:

$$W(C) = \rho_C(\rho_N(b_N - c_N) + \rho_A(b_A - c_A)) \tag{1}$$
$$W(D) = \rho_D(\rho_N b_N + \rho_A b_A) \tag{2}$$

In equations (1) and (2), the expected payoffs are calculated from the product of the remained resources and the frequencies of the agents in each state.

Let $W_s(C)$ and $W_s(D)$ denote the expected systemic payoffs of All-C and All-D strategies respectively. The systemic payoffs of the strategies can be expressed as follows:

$$W_s(C) = W(C) + wW(D) \tag{3}$$
$$W_s(D) = W(D) + wW(C) \tag{4}$$

In equations (3) and (4), the coefficient w represents the connection weight between the agents. The systemic payoff mechanism uses the connection weight to discount the opponent's payoff by the weight w when the agents calculate the systemic payoff. The agents maximize the systemic payoff which involves the opponent's payoff.

We denote a expected payoff of the entire network by $\overline{W}$. The expected payoff is calculated by summing up all systemic payoffs of the strategies. The frequency of normal agents increases if the repairing is successful otherwise decreases. The dynamics of the self-repairing network can be expressed as follows:

$$\frac{d\rho_N}{dt} = \alpha\rho_N\rho_A\rho_C + \beta\rho_A^2\rho_C - (1-\alpha)\rho_N^2\rho_C - (1-\beta)\rho_A\rho_N\rho_C - \lambda\rho_N \tag{5}$$
$$\frac{d\rho_C}{dt} = \frac{(W_s(C) - \overline{W})\rho_C - (W_s(D) - \overline{W})\rho_D}{\overline{W}} \tag{6}$$

The equation (5) represents the interactions among the agents. Each term except the last term consists of the repair interaction between two agents. The frequency ρ_C indicates the repair probability of the agent. The final term represents the decrease of the normal agents by spontaneous failure. The equation (6) represents frequency dynamics of the strategies. The frequency of All-C strategy grows to large if its systemic payoff $W_s(C)$ is greater than the averaged payoff $\overline{W}$.

3.2 Relation between Systemic Payoff and Kin Selection

We reveal that the systemic payoff involving the connection weight is a similar model of kin selection. In kin selection, a relatedness among individuals is determined according to a parent-child relationship. Specifically, the coefficient of the relatedness among the individuals can be calculated with the probability how much they share same genes [3,4].

The systemic payoff is a cooperation mechanism of the agents that was proposed to make the selfish agents cooperate with each other [5,6]. It only focuses on strength of the connection among the agent instead of the blood relationship. In the systemic payoff, we are able to give any connection weight to the agents.

Hamilton's rule [3,4] of kin selection explains that the evolution of cooperation among individuals emerges if the coefficient of the relatedness is greater than the ratio of the benefit and cost. The payoff of kin selection obviously includes the effect of cooperation from other individuals. However, the benefit of cooperation is discounted by the coefficient r of the relatedness among the individuals. Hamilton's rule is simply expressed as follows:

$$rB - C > 0 \tag{7}$$

The B and C represent the benefit of cooperation and cost of defection respectively. We formulate the systemic payoff of All-C strategy as Hamilton's rule. The frequency of All-C increases if the total benefit of the agent and its neighbors is higher than the cost. This condition can be expressed as follows:

$$\begin{aligned} \rho_C(\rho_N b_N + \rho_A b_{[A}) - \rho_C(\rho_N c_N + \rho_A c_A) + wW(D) > \rho_C(\rho_N b_N + \rho_A b_{[A}) \\ w(\rho_D(\rho_N b_N + \rho_A b_A)) - \rho_C(\rho_N c_N + \rho_A c_A) > 0 \end{aligned} \tag{8}$$

We denote the $B' = \rho_D(\rho_N b_N + \rho_A b_A)$ and $C' = \rho_C(\rho_N c_N + \rho_A c_A)$ respectively. Thus, we obtain

$$wB' - C' > 0 \tag{9}$$

The frequencies of the strategies and the states of the agents change dynamically. The frequency of the All-C strategy become large when equation (9) satisfies this condition.

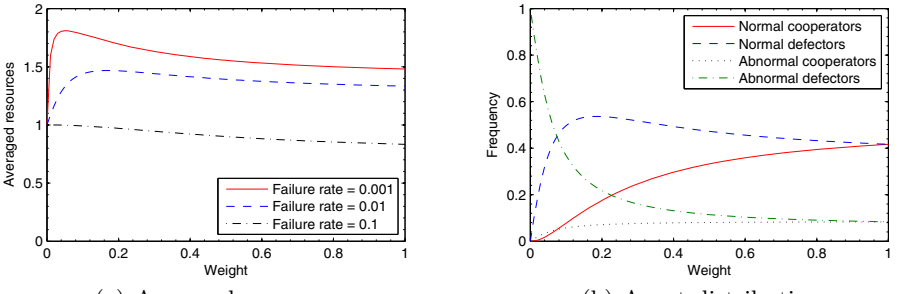
3.3 Performance Evaluations by Numerical Simulations

We reveal the effectiveness of the connection weight for the performance of the self-repairing network. This paper examines the performance of the self-repairing network by varying the connection weight among the agents. We use the averaged resources in the self-repairing network as a criterion of the performance. We compare the performance in the steady state of the network by the numerical simulations. The table 1 shows the parameters for the numerical simulations.

Fig.1 shows the performance and the agents distribution in the steady state of the dynamics of the self-repairing network. Fig.1(a) shows the averaged resources for each failure rate. The averaged payoff of each failure rate has the peak when the connection weight is changed. The connection weight and the environments affect the peak of the averaged resources. Thus, we need to set the appropriate connection weight to the agents in order to elicit the network performance.

Table 1. Parameters for numerical simulations

Parameter	Name	Value
α	Repair success rate by normal agents	1.0
β	Repair success rate by abnormal agents	0.1
b_N	Available resources of normal agents	2.0
c_N	Repair cost of normal agents	1.0
b_A	Available resources of abnormal agents	1.0
c_A	Repair cost of abnormal agents	1.0
λ	Failure rate	0.001, 0.01, 0.1
r	Connection weight	0.0 - 1.0



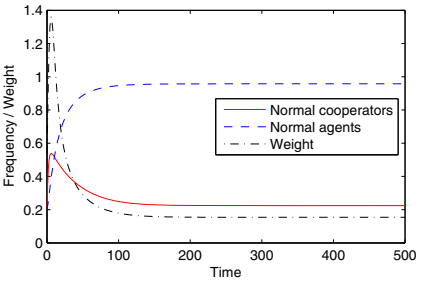
(a) Averaged resources (b) Agent distribution

Fig. 1. Averaged resources of numerical simulations with varying connection weight among agents. The numerical simulations use the parameters shown in Table 1. (a) The averaged resources when the connection weight and the failure rate are varied. (b) The frequency of agent distribution when the failure rate is 0.01.

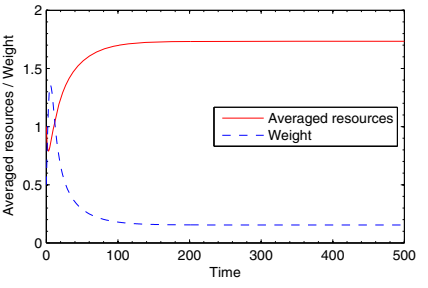
Fig. 1(b) shows the frequency of the agent distribution in the steady state when the failure rate is 0.01. This figure shows that the high connection weights are capable of increasing the frequency of the normal agents. The performance decreases when the connection weight is larger than the peak's weight of the averaged resources (Fig. 1(a)). The normal cooperators grow by raising the connection weight. However, the high connection weights may lead the network to the bad performance than the peak performance because many agents pay the cost for repairing. Thus, the connection weights have to be controlled appropriately to prevent the bad performance.

3.4 Adaptive Control Technique of Connection Weight

We propose the adaptive control technique for the connection weight among the agents. In the previous section, we pointed out that the connection weight of the agents needed to be controlled appropriately. The proper connection weight is different by the environments. The main idea of updating the connection weight is that agents increase the weight if the network performance is bad, otherwise



(a) Agent distribution



(b) Averaged resources

Fig. 2. Time evolution of the self-repairing network with the adaptive control of the connection weight among agents. The simulation parameters are used in Table 1. The failure rate is 0.01.

they decrease the value. Our proposed technique has the simple update rule. Each agent knows the maximum resources value b_N. The agents update their connection weight by comparing their averaged resources in the neighbors and the maximum resources value.

We denote the connection weight by $w(t)$ at time t. The update rule of the connection weight is expressed as follows:

$$\frac{dw}{dt} = \frac{b_N - \overline{W}}{b_N} \tag{10}$$

Therefore, the dynamics of the self-repairing network consists of three equations (5), (6) and (10). We carry out the numerical simulations of the self-repairing network with the adaptive control for the connection weight. We calculate the time evolution of the equations by Runge-Kutta method [11] which is an numerical analysis algorithm for solving ordinary differential equations.

Fig. 2 shows the time evolution of the self-repairing network with the adaptive control of the connection weight. In the transient phase, the connection weight rapidly increases, then it is adjusted to the proper value in the steady state. The normal agents increases in the steady state of the network (Fig. 2(a)). The averaged resources also increase sufficiently in the steady state. The connection weight of the agents is controlled appropriately (Fig. 2(b)). Furthermore, the averaged resources increases to the large value when the connection weight is settled to the fixed value. Our proposed technique succeeded in controlling the connection weight among the agents in numerical simulations.

4 Simulations

4.1 Agent Simulation Model

We carry out the agent simulations of the self-repairing network. In the agent simulations, we examine both technique of the connection weight: fixed and adaptive control.

We consider the self-repairing network in the dynamic environments in which the failure rate changes dynamically. We assume that the failure rate is one of the factors of the environments. Analyzing the self-repairing network in the dynamic environments is important because the environments of the actual computer networks change dynamically. The normal agents become abnormal with the failure rate that oscillates like a sin wave with the cycle 300. The agents need to change the action to adapt the fluctuations of the dynamic environment. Because they would lead them to the absorb state where all of them are abnormal if no agents repair them.

The numerical simulations ignored the particular network structures. The actual computer networks have various network structures. In agent simulations, we apply a particular network structure to the self-repairing network. This paper uses the $L \times L$ square lattice network, because its regularity makes us study deeply than other complex network structures. Each agent are placed on each cell and connected with eight neighbor agents. The interactions of the agents are restricted only eight neighbors.

We consider the simpler payoff of the agents than numerical simulations because the network structure is more complex than the differential equations model. However, the agents calculate their own payoff with the systemic payoff. Each agent has the maximum available resources R_{max}. The agents consume their resources only R_r by repairing. The agents assign their remained resources as the available resources to their own tasks. The remained resources of the abnormal agents are always evaluated as empty because they do not work well due to their state.

The agents need to change their strategies to adapt to the environments. The agents update their strategies to the strategies that earn the highest score in the neighbors. The strategy update of the agents is done with strategy update cycle S. The strategy update error occurs when the agents update their strategies. The agents adopt the other strategies with a strategy update error rate μ. This mechanism contributes to prevent the local minima of the network.

Table 2 shows the parameters of the agent simulations. Our computer simulations in the dynamic environments use the parameters shown in Table 2.

4.2 Performance Evaluations in Dynamic Environments

We use the maximum failure rate shown in Table 2. Fig. 3(a) shows that the averaged resources of the fixed connection weight increases as the connection weight rises to the large value. This tendency is similar to the results of the numerical simulations. The adaptive control of the connection weight succeeds in adjusting its value and shows the peak performance for each failure rate. The peak performances of the adaptive control technique outperform the fixed weight's results.

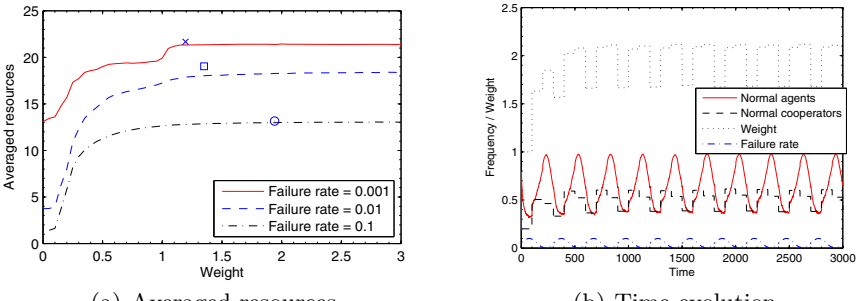
Fig. 3(b) shows the time evolution of the adaptive control technique in the dynamic environments. The proposed technique is capable of adjusting the connection weight dynamically to adapt the changes of the environments. The connection weight of the agents becomes the large value when the frequency of the

Table 2. Parameters for agent simulations

Parameter	Name	Value
T_a	Step	10000
N_t	Number of trial	15
$L \times L$	Number of total agents	2500
$N(0)$	Number of normal agents at initial step	500
All-C	Number of repair agents at initial step	500
α	Repair success rate by normal agents	1.0
β	Repair success rate by abnormal agents	0.1
R_{max}	Maximum resources	25
R_r	Repair resources	1
λ	Failure rate	0.001, 0.01, 0.1
S	Strategy update cycle	100
μ	Strategy update error rate	0.01
w	connection weight	0.0 - 1.3

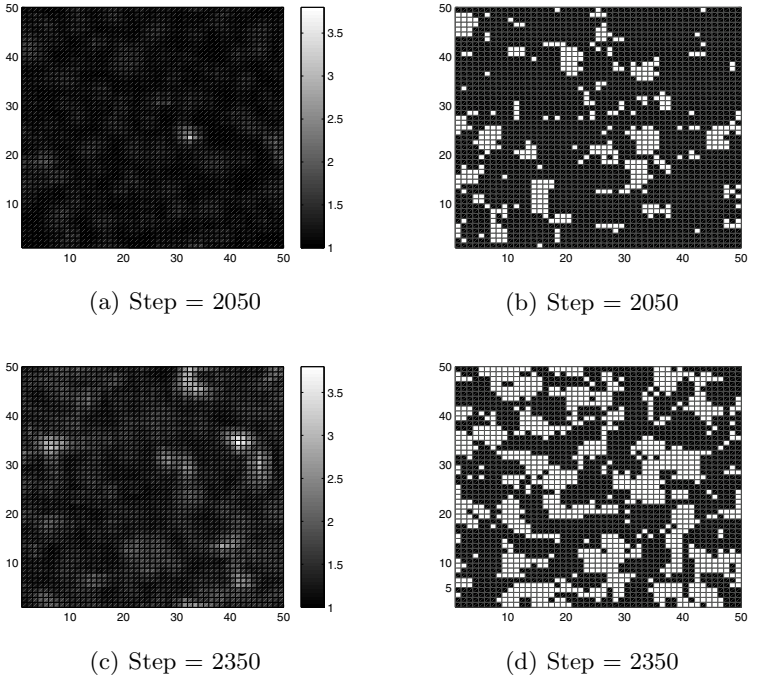
cooperators is the small value. Increasing the connection weight brings out cooperation among the agents, consequently the repair agents appear. The proper connection weight contributes to maintain cooperation among the agents. Therefore, these simulation results are in agreement with the differential equations model presented in the previous section.

Fig. 4 shows the spatial distribution of the strategies and the connection weights. In Fig. 4(a), almost the cells are occupied by black color. Almost the agents set their connection weight to the low values. The agents need not change the connection weight to the high value. The agents do not have to cooperate



(a) Averaged resources (b) Time evolution

Fig. 3. Averaged resources of the self-repairing network in the dynamic environments. (a) The averaged resources are varied by the connection weight. The averaged resources are averaged by the number of the trial counts. The cross, rectangle and circle markers indicate the averaged resources of the adaptive control of the connection weight where the failure rate 0.001, 0.01 and 0.1 respectively. The horizontal point of each marker corresponds to the averaged connection weights. (b) Time evolution of frequency of agent and connection weight in the dynamic environment for the failure rate 0.01.



(a) Step = 2050 (b) Step = 2050

(c) Step = 2350 (d) Step = 2350

Fig. 4. Spatial distribution of the connection weight and strategies in the dynamic environment. (a) and (c) show the spatial distribution of the connection weight. The connection weight is indicated by the color bar. (b) and (d) show the spatial distribution of strategies. The black and white cell respectively represent All-D and All-C strategy agents. The failure rate is 0.01.

with other agents in this situation because the fraction of abnormal agents is small. The unnecessary repairing will waste their resources.

On the other hand, the agents with the high connection weight exist in the environment with the high failure rate (Fig. 4(c)). In this situation, the possibility that the agents become abnormal by the failure rises to the high level (Fig.3(b)). The connection weight of the agents grows to the large value, and then cooperation among the agents emerges (Fig. 4(d)). Because of the systemic payoffs of the agents decrease if the abnormal agents exist in the neighborhood. In this case, the agents change their connection weight to the large value. Then, the agents choosing the All-C strategy obtain the higher remained resources than the All-D strategy agents. Therefore, the altruistic agents repairing other agents appear in the dynamic environment with the high failure rate.

The agents organize the clusters (Fig. 4(c)) but some clusters are independent of the spatial distribution of the strategies (Fig. 4(d)). The high connection weight clusters always do not correspond to the clusters of the altruistic agents. Some agents in the clusters with the high connection weight choose the selfish strategy. In this situation, though the connection weight is high, some normal

agents exist in the neighborhood. From these results, the agents control their connection weight spatially and temporally in the dynamic environments.

5 Discussions

From the numerical and agent simulations, the proposed technique demonstrates that the agents can adapt to the changes of the environments spatially and temporally. The technique are able to contribute cooperation between the agents by changing their connection weight.

We think that systemic payoff with the connection weight would help us to design and construct autonomous distributed systems in computer networks. We considered the systemic payoff that involves the connection weight representing the strength of the relationship among the agents. In simulations, cooperation among the agents emerges where the connection weight is sufficiently large. The averaged resources of the agents are kept enough in this situation. The degree of cooperation is controlled by the parameter of the connection weight. The proper connection weight enables the agents cooperate with other agents in terms of the repairing in the self-repairing network. Therefore, the systemic payoff with the proper connection weight would lead the selfish agents to cooperation for improving the network performance in computer networks.

Furthermore, the adaptive control technique for the connection weight would help us for designing autonomous distributed systems in the dynamic environments. The agent simulations demonstrated that the proposed technique dynamically change the degree of cooperation between agents in computer networks. The agents may choose the fixed connection weight in the static environments. However, environments of computer networks are unpredictable. The unpredictable events such as an infection of computer viruses occur in computer networks. The possibility of the infection by malicious programs will be changed in the dynamic environments. The simulations demonstrated that the agents dynamically change their connection weight to adapt to the changes of the environments. The high connection weight makes the selfish agents cooperate with others in the risky environments. After changing their connection weight, the repair rate of the agents is controlled appropriately. Therefore, the adaptive control technique for the connection weight would become a part of the design method for autonomous distributed systems in computer networks.

6 Conclusions

This paper investigated the relationship between the systemic payoff and kin selection. We considered the systemic payoff which involves connection weight representing the strength of the relationship among the agents. We revealed that the systemic payoff was expressed as Hamilton's rule. This result states that the growth of cooperators emerges when its payoff is greater than the cost. Then, we studied that the connection weight of the agents affects the performance

of the self-repairing network by numerical simulations. We proposed the adaptive control technique that dynamically changes the connection weight of the agents. In the proposed technique, each agent controls their connection weight by comparing the maximum resources and the averaged resources. The proposed technique enables the agents to change their connection weight appropriately in the autonomous distributed systems under dynamic environments.

Acknowledgement

This work was supported by the Global COE Program "Frontiers of Intelligent Sensing," from the ministry of Education, Culture, Sports, Science and Technology.

References

1. Koutsoupias, E., Papadimitriou, C.H.: Worst-case equilibria. Computer Science Review 3, 65–69 (2009)
2. Roughgarden, T., Tardos, E.: How bad is selfish routing? Journal of the ACM 49, 236–259 (2002)
3. Nowak, M.A.: Five rules for the evolution of cooperation. Science 314, 1560–1563 (2006)
4. Brembs, B.: Hamilton's Theory. In: Encyclopedia of Genetics, pp. 906–910. Academic Press, London (2001)
5. Oohashi, M., Ishida, Y.: A Game Theoretic Approach to Regulating Mutual Repairing in a Self-Repairing Network. In: Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications, pp. 281–286. Springer, Netherlands (2007)
6. Ishida, Y., Tokumitsu, M.: Asymmetric interactions between cooperators and defectors for controlling self-repairing. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part III. LNCS (LNAI), vol. 5179, pp. 440–447. Springer, Heidelberg (2008)
7. Ishida, Y.: A critical phenomenon in a self-repair network by mutual copying. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3682, pp. 86–92. Springer, Heidelberg (2005)
8. Chen, P.A., Kempe, D.: Altruism, selfishness, and spite in traffic routing. In: EC 2008: Proceedings of the 9th ACM Conference on Electronic Commerce, pp. 140–149. ACM, New York (2008)
9. Doebeli, M., Hauert, C.: Models of cooperation based on the prisoner's dilemma and the snowdrift game. Ecology Letters 8, 748–766 (2005)
10. Weibull, J.W.: Evolutionary Game Theory. The MIT Press, Cambridge (1997)
11. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes in C, 2nd edn. Cambridge University Press, Cambridge (1992)

Finding Routing Shortcuts Using an Internet Coordinate System

François Cantin* and Guy Leduc

University of Liège, Belgium
Research Unit in Networking
{francois.cantin,guy.leduc}@ulg.ac.be

Abstract. Overlay routing is a promising way to improve the quality of service in the Internet but its main drawback is scalability: measuring the characteristics of the paths, exchanging the measurement results between the nodes and computing the best routes in the full mesh overlay network generally imply a high consumption of resources. In this paper, we design the basis of a lightweight self-organising one-hop overlay routing mechanism improving the latencies: we define criteria that rely on the information provided by an Internet Coordinate System (ICS) in order to provide a small set of potential one-hop shortcuts for any given path in the network with a small measurement cost. Our best criterion does not guarantee to find the best shortcut for any given path in a network but, even in networks with hundreds or thousands of nodes, it will restrict the search for potential shortcuts to about one or two percent of the total number of nodes.

Keywords: Networking, Overlay routing, Routing shortcuts, Internet coordinate system, Vivaldi.

1 Introduction

Nowadays lots of real-time applications are used in the Internet: voice over IP, online video games, etc. Such applications generally need some QoS (Quality of Service) guarantees and, in particular, low delays between communicating nodes in order to perform correctly. However, since the Internet was not developed with QoS guarantees in mind, the default route between two nodes is not guided by QoS constraints and, in many cases, it is possible to find nodes C that are shortcuts in terms of delays:

$$RTT(A,B) > RTT(A,C) + RTT(C,B) \tag{1}$$

where $RTT(X,Y)$ denotes the RTT (Round Trip Time) between the nodes X and Y, i.e., the time necessary to travel in the network from X to Y and go back from Y to X.

* F. Cantin is a Research Fellow of the Belgian Fund for Research in Industry and Agriculture (FRIA). This work has also been partially supported by the EU under project FP7-Fire ECODE.

C. Bettstetter and C. Gershenson (Eds.): IWSOS 2011, LNCS 6557, pp. 56–67, 2011.

Our objective is to find such shortcuts for any given path in a network in order to obtain smaller delays: if C is a shortcut for the path AB, we intend to use C as relay instead of sending the data directly from A to B. To know exactly which node C is the best shortcut for a given path AB, we must check if the inequality (1) is true for each node C. To be able to do that for any given path AB, it is necessary to measure the RTT of each path in the network: if there are n nodes in the network, we have to measure permanently the RTT of $O(n^2)$ paths. That measurement traffic can be considerable and its impact on the data traffic can be significant. A solution to avoid a too large resource consumption due to measurements is to use estimations of the RTTs instead of measuring them.

We propose to use an Internet Coordinate System (ICS) [10], namely Vivaldi [2], to estimate RTTs in a scalable manner (i.e., without too much measurement overhead) and use this knowledge to find shortcuts in the network. Using the estimations provided by an ICS to detect shortcuts leads to a major problem: by definition, a shortcut situation is a violation of the triangle inequality that disappears after its (imperfect) embedding in a metric feature space [5]. Since there are no shortcuts in the feature space, equation (1) can never be satisfied when estimated RTTs are used instead of real RTTs. To circumvent this problem we propose some detection criteria that combine estimated and measured RTTs.

The remainder of this paper is organized as follows. In section 2, we review existing work related to our study. In section 3, we briefly introduce the concept of an ICS and, in particular, of the ICS named Vivaldi. We also discuss the problem an ICS faces in the Internet due to triangle inequality violations. In section 4 we give two simple shortcuts detection criteria and present the results obtained with these criteria. In section 5 we combine the two simple criteria into one hybrid criterion to get the advantages of both. We finally conclude and discuss future work in section 6.

2 Related Work

Overlay routing is attractive because deploying an overlay requires only the cooperation of the nodes participating in the overlay. It has already been proposed to use overlay routing to improve the performance (and the reliability) of a network: it has been observed in [14] that indirect routing can significantly improve the performance for many paths and, on the basis of these observations, Detour [13] and RON [1] were proposed. The idea of RON is to build a fully connected mesh between the nodes participating in the overlay and to monitor all the paths. If a direct path between two nodes is broken or if it has poor performance it proposes to use relay nodes to reach the destination.

Even if overlay networks seem to be an easy way to improve the performance of the Internet, they are not often used by applications. The main problem is the scalability. Indeed, to obtain the results proposed by RON, it is necessary to measure all the paths and distribute measurements results among the overlay nodes in order to apply a routing algorithm. These operations become costly if a large number of nodes are members of the overlay.

A first approach to improve the scalability of overlay networks consists in eliminating reduntant overlay paths. For example, Nakao et al. [8,9] proposed to do that by using topological information (AS-level topology, etc.).

Another solution consists in reducing the communication overhead generated by the exchanges of the measurements results between all the nodes. In RON this overhead is $O(n^2)$ where n is the number of nodes in the overlay. Sontag et al. [15] proposed a protocol allowing each node to find an optimal one-hop path to any other node with a communication overhead $O(n^{1.5})$.

The last way to improve the scalability of overlay routing is to reduce the measurement overhead. Since the objective is to change traffic routes, we suppose that measurements must be done quite frequently to have accurate information about the state of the network. To circumvent this problem Gummadi et al. [3] proposed to route through random relay nodes instead of doing measurements and they observed that it is sufficient to ensure reliability. However, Sontag et al. [15] observed that it is not sufficient to find good alternative paths considering particular metrics like latency. Recently, Lumezanu et al. [7] proposed to use an ICS to detect one-hop shortcuts in a network. Since these situations cannot be reproduced by the estimations provided by the ICS, their idea consists in using estimation errors to find paths that are edges of one-hop shortcuts. We have also explored that solution at the beginning of our work but we gave up because our observations showed that the impact of one-hop shortcuts on an ICS vary according to which shortcut's edges are measured by the ICS (to compute the estimations) and according to the existence of other shortcuts in the network. We have investigated other ways [4,6] to detect one-hop shortcuts by observing the behaviour of the ICS over time instead of computing the estimation errors at a given time. The results obtained are satisfactory but these detection methods are quite heavy to deploy: they need constant collection of data and the processing of these data has a cost. Since the detection of one-hop shortcuts by analysing the behaviour of an ICS seems difficult we propose to use the estimations provided by the ICS to estimate whether or not some node C is a shortcut for a given path AB.

3 Internet Coordinate Systems

Internet Coordinate Systems (ICS) have been developed to predict end-to-end network latency between all pairs of hosts in a population without extensive measurements. Several prediction algorithms have been proposed, e.g., [10,2]. In such systems, a coordinate vector (or coordinates in short) in a metric space is computed for each node by doing measurements between this node and a few other nodes called neighbors or reference points (also called landmarks). When each node has its own coordinates, the RTT for a given pair of nodes can be estimated by computing the distance in the metric space between their coordinates.

3.1 Vivaldi

Vivaldi [2] is a popular fully distributed ICS that does not require a particular network infrastructure. In Vivaldi, each node computes its own coordinates by doing measurements with a few (typically 32) other nodes called its neighbors. If m denotes the number of neighbors chosen by each node, the measurement cost of Vivaldi in a network containing n nodes is $O(n \times m)$. Since m is a small value compared to n, this is better than the measurement cost $O(n^2)$ obtained if all paths must be measured.

Multiple coordinate spaces have been proposed: multi-dimentional Euclidean spaces, spherical coordinates, etc. For this work, we use a 10-dimensional Euclidean space where each node computes its coordinates by doing measurements with 32 neighbors. Following the results of [2] this will ensure us quite reliable coordinates.

3.2 Triangle Inequality Violations

Consider three nodes A, B and C with $RTT(A, B) = 50ms$, $RTT(A, C) = 10ms$ and $RTT(C, B) = 20ms$. There is a TIV (Triangle Inequality Violation) between these nodes because $RTT(A, C) + RTT(C, B) < RTT(A, B)$. By definition, if a node C violates the triangle inequality with a path AB, it means that C is a one-hop shortcut for the path AB. Studies [17] have shown that TIVs are common in the Internet and this is a real problem for an ICS because the triangle inequality must hold in metric spaces. Consequently, TIV situations will inevitably generate estimation errors. Indeed, to try and represent a TIV situation in a metric space, the ICS will have to under-estimate and/or over-estimate the RTT of some paths. It becomes a problem for us if we want to use the estimations to find one-hop shortcuts, because it is impossible to find three nodes such that

$$EST(A, B) > EST(A, C) + EST(C, B) \tag{2}$$

where EST(X,Y) is the estimated RTT between the nodes X and Y.

4 Two Basic One-Hop Shortcut Detection Criteria

In the previous section, we have seen that using only the estimations provided by an ICS to find the one-hop shortcuts in a network is impossible. So, we have to combine estimations with measurements in order to obtain a shortcut detection criterion. In addition to the estimated RTT of each path in the network, we consider that we can obtain the following measurement results. First, if we look for a shortcut for the path AB, we assume that $RTT(A, B)$ can be measured. Secondly, we assume that we can obtain Vivaldi's measurement results between the nodes and their neighbors.

Given these data, for a given path AB, we want to find criteria that provide a set of C nodes that are probably one-hop shortcuts for that path. As such criteria can provide a potentially large set of nodes, we need also a way to rank the C nodes in order to find the best shortcuts as fast as possible.

4.1 Detection Criteria Definitions

In [7], Lumezanu et al. stated that, if a node C violates the triangle inequality with a path AB, $EST(A, B)$ is an under-estimation of $RTT(A, B)$. We observed this too: generally, more than 80% of the paths for which there exists at least one (significant) shortcut are under-estimated by the ICS. The following criteria are based on that observation. Indeed, if the estimation of the alternative path is reliable and if the path AB is significantly under-estimated, we will restore the TIV by replacing $EST(A, B)$ by $RTT(A, B)$ in (2).

Estimation Detection Criterion (EDC). is our first criterion. To decide if a node C is a shortcut for a path AB, this criterion compares the measured RTT of the direct path between A and B and the estimated RTT of the alternative path using C as relay. Formally, a node C is considered as a shortcut for the path AB if

$$RTT(A, B) > EST(A, C) + EST(C, B) \tag{3}$$

The potential problem with that criterion is that it uses the values of the estimations provided by the ICS as if there were no estimation error. However, we know that there are estimation errors and, in particular, that these errors cannot be avoided if node C is a shortcut for the path AB. So, using the exact values of the estimated RTTs to find shortcuts is not necessarily a good idea.

Approximation Detection Criterion (ADC). is our second criterion. For a path AB and a node C, we define C_A (resp. C_B) as C's nearest node among A's (resp. B's) Vivaldi neighbors according to the estimated RTTs. Since A and C_A (resp. B and C_B) are neighbors, we assume that $RTT(A, C_A)$ (resp. $RTT(B, C_B)$) is known and can be used by the criterion to approximate the RTT of the alternative path: a node C is considered as a shortcut for the path AB if,

$$RTT(A, B) > RTT(A, C_A) + RTT(C_B, B) \tag{4}$$

4.2 Ranking of the Detected C Nodes

We have two criteria which, for a given path AB, are able to return a set of C nodes that are probably shortcuts for that path. The problem with such criteria is that they do not provide a set of nodes containing only the best shortcuts: they provide a possibly large set of nodes containing nodes that are important shortcuts, nodes that are less important shortcuts and even nodes that are not shortcuts (detection errors). So, we need a way to rank the C nodes of a set in order to find quickly and easily the best shortcuts in that set. Since we want to find the node C providing the smallest RTT for a path between A and B, we will rank the C nodes by order of provided gain. For a path AB, the *absolute gain* (G_a) and the *relative gain* (G_r) provided by a node C are

$$G_a = RTT(A, B) - (RTT(A, C) + RTT(C, B)) \qquad G_r = \frac{G_a}{RTT(A, B)}$$

If C is a shortcut for the path AB, then G_a and G_r will have positive values and the most interesting shortcut is the one that provides the highest value for these parameters. However, we cannot compute G_a and G_r for all C nodes. Indeed, generally, we do not know the real RTT of the alternative path that uses node C: we only have Vivaldi's estimations for that path. As we have used an estimation/approximation for the RTT of the alternative path in the shortcut detection criteria, we will also use that estimation/approximation in the ranking criteria. The values used to rank the C nodes of a set will be denoted *estimated absolute gain* (EG_a) and *estimated relative gain* (EG_r). The definitions of these values depend on the shortcut detection criterion used to obtain the set of C nodes:

$$EG_a = RTT(A,B) - (VAL(A,C) + VAL(C,B)) \qquad EG_r = \frac{EG_a}{RTT(A,B)} \quad (5)$$

where $VAL(X,Y)$ is the value used by the detection criterion to estimate the RTT of the path XY.

For a path AB, we will rank the C nodes of the set selected by a shortcut detection criterion in decreasing order of their estimated gain. If the nodes with the highest estimated gains are also those with the highest (real) gains then we will find the nodes providing the most interesting shortcuts in the top of the ranking.

4.3 Performance Evaluation

To model Internet latency, we used three delay matrices containing results of measurements performed in real networks : two publicly available data sets, the P2PSim data (1740 nodes) [11] and the Meridian data (2500 nodes) [16], and a data set we obtained by doing measurements between 180 nodes on Planetlab [12]. In these matrices, the percentage of paths for which there exists at least one shortcut is respectively 86%, 97% and 67%. Since a shortcut is not necessary useful[1], we define an *interesting shortcut* as a shortcut that provides at least an absolute gain of 10 ms and a relative gain of 10%. The percentage of paths for which there exists at least an interesting shortcut in our matrices is respectively 43%, 83% and 16%. So, searching shortcuts in the networks modelled by these matrices can provide an improvement in terms of delays for many paths.

We have simulated the behaviour of Vivaldi on these three networks by using the P2PSim [11] discrete-event simulator. Each node has computed its coordinates in a 10-dimensional Euclidean space by doing measurements with 32 neighbors. Then, we simply applied our detection criteria using the estimated delay matrices computed with the coordinates obtained at the end of the simulations of Vivaldi. We will now evaluate the quality of the sets of detected nodes provided by our criteria.

[1] For example, for a path AB such that $RTT(A,B) = 100\,ms$, a node C such that $RTT(A,C) + RTT(C,B) = 99\,ms$ is a shortcut that provides an absolute gain of $1\,ms$ and a relative gain of 1%. Since using C as relay for sending data from A to B will add an additional forwarding delay, such shortcuts are useless in practice.

Table 1. EDC and ADC shortcut detection results

	EDC			ADC		
	TPR	ITPR	FPR	TPR	ITPR	FPR
P2PSim	53%	83%	2%	65%	84%	9%
Meridian	54%	64%	9%	70%	76%	25%
Planetlab	37%	75%	1%	60%	81%	5%

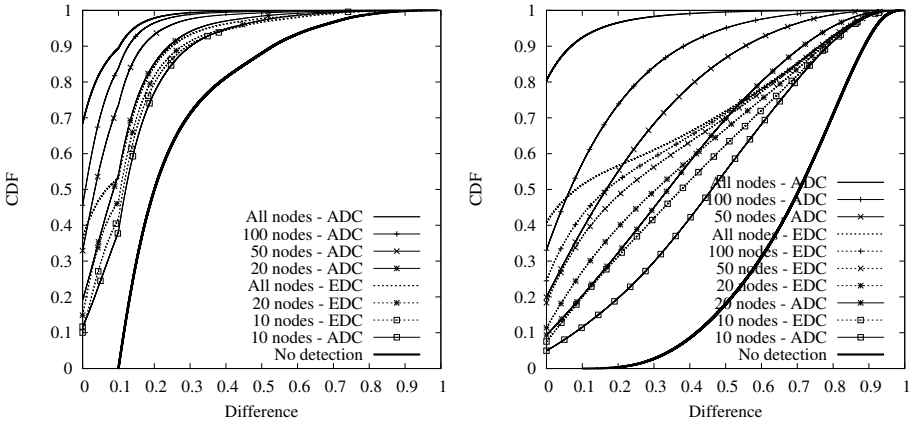
Shortcut detection. To evaluate the performance of our detection criteria, we first use the classical true positive rate and false positive rate indicators. For a path AB, a good shortcut detection criterion must detect a node C as a shortcut if it is a shortcut for the path AB (i.e., if it is a positive) and must reject a node C if it is not a shortcut for the path AB (i.e., if it is a negative). The percentage of positives detected as shortcuts is the *true positive rate* (TPR) and the percentage of negatives detected as shortcuts is the *false positive rate* (FPR). We also define the *interesting true positive rate* (ITPR) as the percentage of interesting shortcuts detected as shortcuts by the criterion. A good detection criterion must provide a high (I)TPR and a low FPR.

The true positive rates and false positive rates obtained with our criteria are given in table 1. We see that the percentage of interesting shortcuts detected as shortcuts (ITPR) is good in most of the cases for both criteria. Furthermore, the percentage of non-shortcuts detected as shortcuts (FPR) is generally quite low. So, these results are satisfactory and, considering these results, EDC seems to perform better than ADC: although ADC is always able to detect slightly more shortcuts than EDC, it also gives more false positives.

Detection of the best shortcuts. Being able to detect lots of the shortcuts in a network is one thing, but what matters most is to detect the most interesting shortcuts (those that provide the most important gain). Considering only the paths for which there exists at least one interesting shortcut, the percentage of paths for which the most interesting shortcut is detected in the matrices P2PSim, Meridian and Planetlab is respectively 36%, 41% and 49% with the EDC criterion and 68%, 80% and 70% with the ADC criterion.

Regarding those results ADC seems to be a better criterion than EDC. Indeed, EDC is able to find the best shortcut for 40% of the paths (on average) while the ADC is able to find the best shortcut for more than 70% of the paths in each matrix. However, we must perhaps moderate our conclusion. Firstly because ADC returns large sets of C nodes (including a non-negligible number of false positives) compared to EDC. At the limit, a criterion that detects as shortcut all C nodes will obviously detect the best shortcut for each path but is completely useless. So, if we choose to use ADC, we absolutely need a criterion[2] to rank the

[2] Such criterion can also be useful for EDC because, even if the sets of C nodes are generally smaller than those returned by ADC, they can contain tens or hundreds of nodes.



(a) P2PSim (b) Meridian

Fig. 1. Comparison of EDC and ADC: Difference of G_r between the best shortcut and the best detected shortcut

C nodes of a set in order to keep only a subset of the nodes. Moreover, EDC can give better results than we think. Indeed, even if a criterion cannot find the best shortcut for a path, it may be able to find another shortcut that provides almost the same gain. We will investigate that during the evaluation of the quality of the ranking of the C nodes.

Ranking of the detected nodes. For a given matrix and a given detection criterion, we proposed in section 4.2 to rank the C nodes of each path on the basis of the EG_r they provide. We will now evaluate if this gives a ranking with the C nodes providing the best G_r in the first positions. For this evaluation, we only consider the paths for which there exists at least one interesting shortcut.

To do this evaluation, we consider for each path that only the first k C nodes of the ranking are detected by the criterion (for several values of the parameter k). For these subsets of C nodes we compute the difference between the G_r provided by the best existing shortcut and the G_r provided by the best shortcut in the subset. We thus obtain one value for each path and we build the CDF (Cumulative Distribution Function) of these values. The CDFs obtained with different values of the parameter k are given in figure 1.

The graphs named "no detection" in figures 4.3 and 4.3 give the CDF of the G_r provided by the best existing shortcut for each path of the matrix. Indeed, if there is no detection criterion applied, there is no shortcut detected and the difference between the G_r provided by the best existing shortcut and the G_r provided by the best detected shortcut is the G_r provided by the best existing shortcut. By applying one shortcut detection criterion, we will detect some shortcuts and, thus, reduce that difference for some paths. Since the computed difference is smaller for more paths, the CDF will rise faster on the graphs.

The graphs named "all nodes - XDC" in the subfigures of figure 1 give the CDF computed by considering all the nodes selected by the shortcut detection criterion XDC (ADC or EDC). This is equivalent to using $k = \infty$. These are the best results that the given detection criterion applied on the given matrix can provide. We can see that ADC still gives better results than EDC considering those graphs. Indeed, with ADC, there are only a small part of the paths for which the difference between the G_r provided by the best existing shortcut and the G_r provided by the best detected shortcut is bigger than 0.2. That means that, for a small part of the paths AB, it is still possible to find another shortcut C that would improve by more than 20% of $RTT(A, B)$ the gain provided by the alternative path proposed by our shortcut detection criterion. This difference is generally bigger with EDC.

Let us see what is the situation if we keep only the first nodes of the rankings. The graphs named "k nodes - XDC" in the subfigures of figure 1 give the CDF computed by considering as detected only the first k nodes of the rankings obtained by using the shortcut detection criterion XDC (ADC or EDC). The first thing we see is that even if we take only a few nodes in the ranking (e.g., 10 nodes), we obtain already a good improvement compared to the situation whithout shortcut detection. We also see that ADC gives better results than EDC only if we keep a sufficient number of nodes: more than 50 nodes for Meridian, more than 20 nodes for P2PSim and more than 5 nodes for Planetlab[3]. Moreover, if we keep a sufficient number of nodes (100 nodes for Meridian, 20 nodes for P2PSim and 10 nodes for Planetlab), we obtain a result with ADC that is better than what we can obtain by considering all the nodes with EDC. The number of nodes to keep to obtain good results may seem important for Meridian but it represents only 4% of the total number of nodes.

Given those results we can conclude that, with ADC, when considering only 5% of the total number of nodes in each matrix (that represents 125 nodes for Meridian, 87 nodes for P2PSim and 9 nodes for Planetlab), we are able to provide a significant improvement of the RTT for lots of paths for which there exists at least one interesting shortcut.

5 Hybrid One-Hop Shortcut Detection Criterion

It is possible to obtain better results than those obtained with ADC. Indeed, if it is impossible to find some A's (resp. B's) Vivaldi neighbors near C, the approximation of $RTT(A, C)$ (resp. $RTT(C, B)$) by $RTT(A, C_A)$ (resp. $RTT(C_B, B)$) can be very bad. In such case, using the EDC criterion can provide more reliable detection results even if there are estimation errors. So, we define a *Hybrid Detection Criterion* (HDC) by combining our two basic criteria in order to exploit their advantages.

[3] Since there are only 180 nodes in the Planetlab matrix, the sets of C nodes returned by the criteria are quite small and keeping all the detected nodes is not really a problem. So, the quality of the ranking is less important for that matrix and we will not show the graphs here.

5.1 Criterion Definition

Formally, let C_A (resp. C_B) be C's nearest node among A's (resp. B's) Vivaldi neighbors according to the estimated RTTs. We define

$$VAL(A,C) = \begin{cases} RTT(A,C_A) \text{ if } EST(C_A,C) < threshold \\ EST(A,C) \quad \text{otherwise} \end{cases}$$

$$VAL(C,B) = \begin{cases} RTT(C_B,B) \text{ if } EST(C_B,C) < threshold \\ EST(C,B) \quad \text{otherwise} \end{cases}$$

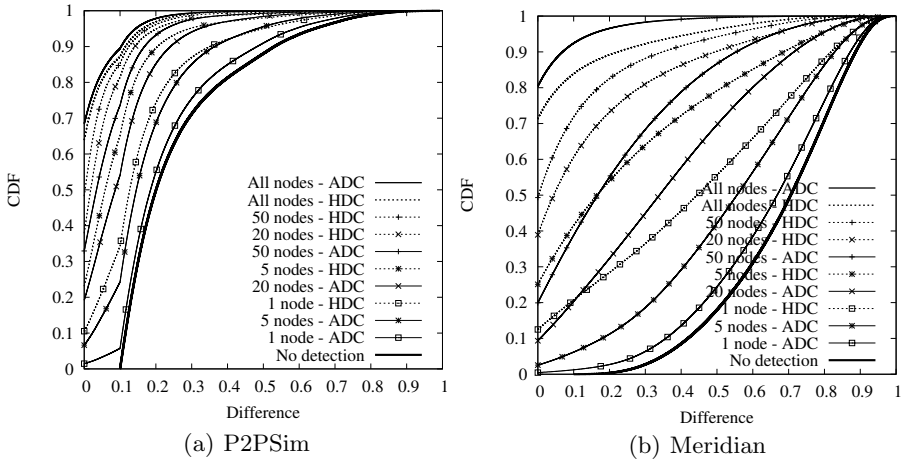
where *threshold* is a value used to decide if C_A (resp. C_B) is sufficiently near C to obtain a quite good approximation of $RTT(A,C)$ (resp. $RTT(C,B)$) by using $RTT(A,C_A)$ (resp. $RTT(C_B,B)$). To test the HDC criterion, we choose *threshold* equal to 10% of $RTT(A,B)$ when we search a shortcut for the path AB. Fine tuning this parameter is part of our future work. Using these definitions, a node C is considered as a shortcut for the path AB if

$$RTT(A,B) > VAL(A,C) + VAL(C,B) \tag{6}$$

To rank the C nodes of the set provided by this criterion for a given path AB, we proceed as described in section 4.2.

5.2 Performance Evaluation

For the evaluation of HDC, we consider only the paths for which there exists at least one interesting shortcut. Let us begin with the detection of the best shortcuts. The percentages of paths for which the most interesting shortcut is detected with the HDC in the matrices P2PSim, Meridian and Planetlab are respectively 64%, 71% and 74%. These results are better than those obtained



(a) P2PSim (b) Meridian

Fig. 2. Comparison of ADC and HDC: Difference of G_r between the best shortcut and the best detected shortcut

with EDC but are worse than those obtained with ADC. So, HDC misses interesting shortcuts that ADC is able to find. Moreover, in figure 2, considering all the nodes detected by the criterion, we can see that ADC is potentially able to provide a better improvement of the latencies in the network than HDC.

However, considering the quality of the ranking of the C nodes, we see in figure 2 (graphs named "k nodes - XDC") that HDC performs a lot better than ADC. Indeed, with HDC, even if we take only the first node of the ranking, we obtain already a significant improvement compared to the situation without shortcut detection. Furthermore, if we only consider the first 5 nodes of a ranking, we obtain better results than if we consider the first 50 nodes of the rankings with ADC. So, with HDC, we can provide a substantial improvement of the RTT for lots of paths by considering only about 2% of the total number of nodes (that represents only 50 nodes for Meridian, 34 nodes for P2PSim and 3 nodes for Planetlab).

6 Conclusion and Future Work

In this article, we showed that, for any given path AB, using only the RTT of that path and the information available in an ICS, it is possible to select a small set of nodes containing very likely an interesting one-hop shortcut (but not necessarily the best one) when shortcuts exist for that path. We obtained the best results with our shortcut detection criterion called HDC. With that criterion we are able to limit the number of potential shortcuts for any given path AB to about one or two percent of the total number of nodes in the network. So, to improve significantly the latency between A and B, we will only have to do measurements between A, B and these few candidate nodes to know if they are really shortcuts and which of them is the best shortcut. These results are encouraging and it is probably possible to tune the HDC threshold to obtain even better results.

Our future work will consist in designing a distributed self-organised one-hop routing mechanism based on the encouraging observations reported so far. The distributed and self-organised aspects of the mechanism are natural since the ICS exhibits these properties and a node A can search shortcuts between itself and a node B simply by collecting the coordinates of the other nodes and some more information about B. We will also have to consider problems like churn (arrival and departure of nodes), incentives (motivate a node to participate as relay [7]), etc. Compared to a solution like RON [1] the traffic generated by the measurements will be significantly reduced with our approach. With RON, since each node has to do RTT measurements with all the other nodes this traffic is $O(n^2)$. With our approach, each node measures only with its m neighbors and this traffic is $O(n \times m)$ (with $m \ll n$). Our approach will also reduce the communication overhead generated by the exchanges of the measurements results. With RON, each node has to send its measurement results to each other nodes. So, the number of messages is $O(n^2)$ and the size of each message is $O(n)$. Consequently, the traffic generated is $O(n^3)$. With our approach this traffic is

only $O(n^2)$: each node will have to send a message to each other nodes (n^2 messages) but each message contains only one coordinate vector. Finally, our approach will be less efficient than RON in terms of quality of the shortcuts proposed but it will generate less traffic in the network.

References

1. Andersen, D., Balakrishnan, H., Kaashoek, F., Morris, R.: Resilient overlay networks. SIGOPS Oper. Syst. Rev. 35(5), 131–145 (2001)
2. Dabek, F., Cox, R., Kaashoek, F., Morris, R.: Vivaldi: A decentralized network coordinate system. In: Proc. of SIGCOMM, Portland, OR, USA (August 2004)
3. Gummadi, K.P., Madhyastha, H.V., Gribble, S.D., Levy, H.M., Wetherall, D.: Improving the reliability of internet paths with one-hop source routing. In: Proceedings of OSDI. USENIX Association, Berkeley (2004)
4. Kaafar, M., Cantin, F., Gueye, B., Leduc, G.: Detecting triangle inequality violations for internet coordinate systems. In: Proc. of Future Networks 2009 Workshop, Dresden, Germany (June 2009)
5. Ledlie, J., Gardner, P., Seltzer, M.I.: Network coordinates in the wild. In: Proc of NSDI, Cambridge, UK (April 2007)
6. Liao, Y., Kaafar, M., Gueye, B., Cantin, F., Geurts, P., Leduc, G.: Detecting triangle inequality violations in internet coordinate systems by supervised learning - work in progress. In: Proc. of Networking 2009, Aachen, Germany (May 2009)
7. Lumezanu, C., Baden, R., Levin, D., Spring, N., Bhattacharjee, B.: Symbiotic relationships in internet routing overlays. In: Proceedings of NSDI, pp. 467–480. USENIX Association, Berkeley (2009)
8. Nakao, A., Peterson, L., Bavier, A.: A routing underlay for overlay networks. In: Proceedings of SIGCOMM, pp. 11–18. ACM, New York (2003)
9. Nakao, A., Peterson, L., Bavier, A.: Scalable routing overlay networks. SIGOPS Oper. Syst. Rev. 40(1), 49–61 (2006)
10. Ng, T.S.E., Zhang, H.: Predicting Internet network distance with coordinates-based approaches. In: Proc. of INFOCOM, New York, NY, USA (June 2002)
11. A simulator for peer-to-peer protocols, `http://www.pdos.lcs.mit.edu/p2psim/index.html`
12. PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services, `http://www.planet-lab.org`
13. Savage, S., Anderson, T., Aggarwal, A., Becker, D., Cardwell, N., Collins, A., Hoffman, E., Snell, J., Vahdat, A., Voelker, G., Zahorjan, J.: Detour: Informed internet routing and transport. IEEE Micro. 19(1), 50–59 (1999)
14. Savage, S., Collins, A., Hoffman, E., Snell, J., Anderson, T.: The end-to-end effects of internet path selection. SIGCOMM Comput. Commun. Rev. 29(4), 289–299 (1999)
15. Sontag, D., Zhang, Y., Phanishayee, A., Andersen, D.G., Karger, D.: Scaling all-pairs overlay routing. In: Proceedings of CoNEXT, Rome, Italy (December 2009)
16. Wong, B., Slivkins, A., Sirer, E.: Meridian: A lightweight network location service without virtual coordinates. In: Proc. of the ACM SIGCOMM (August 2005)
17. Zheng, H., Lua, E.K., Pias, M., Griffin, T.G.: Internet routing policies and round-trip-times. In: Dovrolis, C. (ed.) PAM 2005. LNCS, vol. 3431, pp. 236–250. Springer, Heidelberg (2005)

Cache Capacity Allocation to Overlay Swarms

Ioanna Papafili[1], George D. Stamoulis[1], Frank Lehrieder[2], Benjamin Kleine[2], and Simon Oechsner[2]

[1] Athens University of Economics and Business, Athens, Greece
[2] Julius-Maximilian University of Wuerzburg, Wuerzburg, Germany

Abstract. Peer-to-peer applications generate huge volumes of Internet traffic, thus leading to higher congestion, as well as higher costs for the ISPs particularly due to inter-domain traffic. The traditional traffic management approaches employed by ISPs (e.g. traffic shaping or even throttling) often lead to a deterioration of users' Quality-of-Experience. Previous works have verified that the insertion of ISP-owned Peers (IoPs) can deal effectively with this situation. The mechanism offers caching while exploiting the self-organizing structure of overlays. Thus, it leads to both improved performance for peers and reduced inter-domain traffic costs for ISPs. In this paper, we study how the available IoP bandwidth capacity should be allocated among the various swarms that it can possibly join. In particular, we identify a variety of factors that influence the effectiveness of Swarm Selection and Bandwidth Allocation, and we investigate their impact on the practically relevant example of BitTorrent, primarily by means of simulations.

Keywords: peer-to-peer, ISP-owned Peer (IoP), cost reduction, performance improvement, Swarm Selection, Bandwidth Allocation.

1 Introduction

Dissemination of large content files by means of peer-to-peer (P2P) applications is very popular among Internet users for almost 5 years. However, due to their popularity and the size of the files exchanged, P2P applications generate huge traffic volumes. Furthermore, they are highly self-organized and perform their own overlay traffic management, generally without taking into account the underlying network. Thus, it is possible that some information is brought to a peer from a remote one, while being locally available. Due to the resulting increase of inter-domain traffic caused by P2P applications, ISPs pay increased inter-connection charges. Thus, ISPs employ traffic engineering aiming mainly to reduce traffic volumes on their interconnection links. Traffic optimization performed separately by ISPs and overlays leads the Internet eco-system to a suboptimal situation [1]. Thus, there is need for an incentive compatible mechanism to exploit the nice characteristics of self-organized overlay networks and co-assist ISP's traffic engineering, and lead the system to a *win-win* situation; e.g. leading to both reduced inter-domain traffic charges for ISPs and improved performance for overlay users. This is the objective of Economic Traffic Management (ETM), which is the subject of EU-funded project SmoothIT [2].

C. Bettstetter and C. Gershenson (Eds.): IWSOS 2011, LNCS 6557, pp. 68–80, 2011.

An ETM mechanism that appears to achieve a win-win situation is the insertion of ISP-owned Peer(s), proposed and initially analyzed in [3] for the case of a *single* swarm. The ISP-owned Peer (IoP) is an entity equipped with abundant resources, e.g. bandwidth and storage, managed and controlled by the ISP. Note that the IoP runs the overlay protocol with minor modifications, e.g., due to its high amount of resources, it is allowed to unchoke more than the regular number of peers, and thus participates actively in the overlay, storing content and aiming at subsequently seeding this content. The underlying objective is both to increase the level of traffic locality within an ISP, and thus reduce traffic redundancy on its inter-domain links, and to improve performance experienced by the users of peer-to-peer applications (win-win). For the case of BitTorrent [4], which is henceforth assumed, by exploiting the self-organizing incentive mechanism 'tit-for-tat' and due to its high resources, the IoP will be preferred by regular peers to exchange data with. Furthermore, since the IoP is not managed by an actual user but the ISP, and deals with multiple swarms, certain important functions need to be determined. Namely: *i) how the IoP will discover new swarms to join, ii) how it will decide which one of the known swarms to join, and iii) how it will allocate its resources among the swarms it has joined.*

In this paper, we focus on the Swarm Selection and Bandwidth Allocation procedures on the example of the popular and widely-used BitTorrent protocol. We study the impact of these two procedures on the IoP's performance and investigate how different factors affect them. In fact, in practical cases with dynamically varying conditions the IoP would *adapt* its decisions to the changes of these factors in a self-organized way. We use simulative performance evaluation mainly. Thus, we have performed extensive simulations using the multi-swarm capable SmoothIT Simulator v3.0 [5], which is based on the ProtoPeer platform [6]. Note that the impact of cache insertion has been studied already in the literature (see Section 5) but only for single-swarm scenarios. We are not aware of any research that considers a similar multi-swarm approach and the relevant questions arising, as is done in this work. It should be noted that storing content that is illegally shared by users is problem for ISPs. However, this is generally true for any caching approach, of which there are several commercially offered and used in practice, such as products from Oversi [7] or PeerApp [8]. Thus, here we limit ourselves to technical analysis of the IoP approach.

The remainder of the paper is organized as follows: in Section 2, we describe the IoP insertion mechanism. In Section 3, we investigate Swarm Selection and in Section 4, we study the Bandwidth Allocation. In Section 5, we present a brief survey of related work, while in Section 6 we discuss the evaluation results presented in Sections 3 and 4 and conclude our work.

2 ISP-Owned Peer Insertion

Since we study our topics of interest using the practically relevant example of the BitTorrent protocol, we first briefly describe the most important mechanisms of this overlay. BitTorrent is a file-sharing application for the dissemination of large content files to a large number of peers. All peers sharing the same content file form one overlay, which is called a swarm. Peers download a file with meta-data called torrent; the meta-data includes hash values of the content file and the address of the tracker. The

tracker is a centralized entity responsible for neighbor discovering that keeps a complete list of peers that participate in a swarm. Peers request a list of neighbors from the tracker, and the tracker returns a *random* list of peers, both leechers and seeders. Leechers are peers that have only part of the content file, while seeders are peers that have the complete file. Then data chunks exchange among peers begins. The data exchange follows the rules of the BitTorrent protocol; the most significant ones are the choke algorithm and the rarest first replication.

While the insertion of an ISP-owned (and controlled) peer seems to be an indirect way of an ISP to manage its P2P traffic, it is based on the exploitation of the self-organization incentive mechanism of BitTorrent, namely the choke algorithm [4], to increase its effectiveness. The choke algorithm attempts to optimize peers' download rates by employing a variant of 'tit-for-tat'; peers reciprocate to each other by uploading to peers which upload to them with highest rate. Due to its high amount of resources, the IoP is expected to be unchoked with a higher probability by other peers, and therefore is able to concentrate all data chunks more quickly and then help regular peers acquire them. (Note that an ISP can also decide to restrict uploading from the IoP by non-local peers.) This will lead to a reduction of traffic in the inter-domain links, and of the associated charges of the ISP (win), under tariffs such as those based on incoming inter-domain traffic or on the total such traffic, which are often applied in practical cases. However, our target is also to maintain ("no lose" for the overlay) or improve ("win" for the overlay) the performance experienced by the users. Since the aim of the ISP is to affect as much P2P traffic as possible, the IoP should participate in a set of swarms where it can join and offer its available resources more effectively; thus, two important mechanisms have to be implemented *periodically* by the IoP: the ***Swarm Selection*** and the ***Bandwidth Allocation*** mechanisms.

Swarm Selection is the mechanism that selects those swarms that the IoP will join out of the set of all *known* swarms, so that the IoP has a high impact in terms of inter-domain traffic reduction and peers' performance improvement. The detection of new swarms and the discovery of IoP by new peers joining swarms supported by IoP are out of the scope of this paper. Here, we just assume that the IoP is updated on all new and existing swarms by an external entity also managed by the ISP; the same entity is responsible for helping new peers discover IoP; see [9]. Swarm Selection includes a rating of different swarms based on overlay criteria (e.g. content file size, the numbers of leechers and seeders), underlay criteria (e.g. access bandwidth of local peers in the swarm), or both, and finally the selection of a set of swarms with the highest rating.

Bandwidth Allocation is the mechanism responsible for the efficient distribution of the IoP bandwidth (i.e., data rate) among all swarms that the IoP currently joins according to Swarm Selection. The actual data rate that will be used by the peers of a swarm cannot be a priori decided by the IoP; on the contrary, it depends on the number of peers, their download rate, etc. Bandwidth Allocation implies resource dedication to a swarm for the time interval until the bandwidth allocation algorithm re-runs; beyond that bandwidth is provided to peers inside the same swarm based on the overlay protocol. We investigate three different types of Bandwidth Allocation in this work, *uniform* policy, *proportional* policy (according to the ratio of the number of local leechers to the total swarm size) or finally, *max-min* policy (again according to this ratio), which are presented and studied in Section 4.

3 Swarm Selection

In this section, we provide an analysis of the Swarm Selection mechanism, investigating the impact of the content file size, the number of local leechers and the number of local seeders, individually and in pairs, on IoP's effectiveness. For simplicity we consider a static case with two swarms evolving simultaneously (see below), and investigate the impact of the aforementioned factors in static IoP Swarm Selection; i.e. in deciding which swarm the IoP should join. Since we cannot a priori determine the dynamically-varying values of the number of leechers and the number of seeders in our simulations, we use instead the *static* parameters of mean inter-arrival times of leechers and the mean seeding time of seeders to prescribe the swarm composition and size in the simulation set-up; the parameters mentioned are those that are *observable on-line* and thus can be the basis for the periodic IoP decision making. Note that here we assume that the IoP unchokes remote peers with no restriction.

In order to verify monotonicity of the leechers' and seeders' number w.r.t. the aforementioned static parameters, we used the theoretical model proposed in [10] that extends the fluid model of [11] to include caches. Thus, we calculate average numbers of leechers and seeders in a swarm w.r.t. the file size, the leechers' arrival rate, or the seeding time in steady-state. For a simple scenario of two ASes and two swarms and using default values for the parameters of the model (as given in [10]), we have derived results for cases of different file sizes, mean inter-arrival times or seeding times. Numerical results (which are omitted, for brevity reasons) show that the average number of leechers is (as intuitively expected) increasing with the file size and decreasing with the mean inter-arrival time and the mean seeding time. These justify our selection of the two aforementioned static factors as criteria for Swarm Selection. Below, we first present the simulation setup for our evaluations and subsequently evaluation results derived from these simulations.

Simulation setup. For the evaluation of the Swarm Selection mechanism, we have used the topology of two symmetric ASes, inter-connected via transit inter-AS links to a hub AS, shown in Fig. 1. The tracker and original seeder are connected to the transit hub AS, which has no peers, while the IoP is always inserted in AS1. The topology is very simple but sufficient to serve our purposes; namely to evaluate the impact of the IoP insertion and its mechanisms on inter-AS traffic and peers' download times. Normal peers have an access bandwidth of 2048/128 kB/s down/up (homogeneous), the initial seeder has a capacity of 1280 kB/s (up and down) while the IoP has a capacity of 6400 kB/s (up and down), which is adequately large to reveal the impact of the Swarm Selection procedure.

In the overlay level, we assume two swarms that are specified by the three set-up parameters file size, mean inter-arrival time and mean seeding time. The latter two replace in the configuration of our simulation the number of leechers and the number of seeders, which can be monitored in practice. Peers arrive according to a Poisson process, and after they finish downloading they remain in the swarm and serve as seeders for a time duration that follows the exponential distribution. The tracker and the IoP never leave the swarm. To start the content distribution process properly, we assume the existence of an initial seeder at the beginning of the simulation. This is a peer that acts as the original source of the shared files and leaves the swarm after 1 hour, so that it has no impact on the steady-state of the swarm.

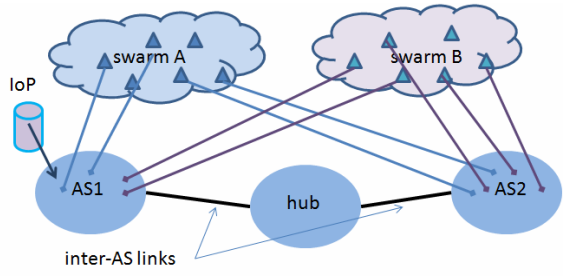


Fig. 1. Two-AS topology

By default, both ASes have some peers that participate only in swarm A, some that participate only in swarm B and some that participate in both swarms. The default file size is 150MB, the leechers' mean inter-arrival time is meanIAT = 100.0s, and the mean seeding time meanST = 600.0s; this results in swarms of about 30 peers concurrently online in steady-state. Such values are common according to recent measurements [12]. To study the impact of the three factors, we tune parameters only for swarm A as reported in Table 1. For swarm B, we always employ these default values. The simulation duration was 3.5 hours, but the results of the first 1.5 hours were ignored as this was considered to be the warm-up phase.

Table 1. Evaluation scenarios for Swarm Selection

Scenario	*A*	*B*	*C*	*D*	*E*	*F*
Modified parameters	File Size: 50 MB	meanIAT: 300.0 s	meanST: 200.0 s	File Size: 50 MB, meanIAT: 300.0 s	File Size: 50 MB, meanST: 200.0 s	meanIAT: 300.0 s, meanST: 200.0 s

Evaluation results for Swarm Selection factors. In Fig. 2, we present results regarding inter-AS inbound traffic of AS for scenarios A, B and C, where only one of the tunable parameters is modified. (All results hereafter are presented with their respective 95% confidence intervals for 10 simulation runs.) We observe that the impact on inbound inter-AS traffic of AS1 is more significant when the IoP has joined a swarm that either i) serves a content file of larger size, or ii) has lower mean inter-arrival times of leechers and, thus, more leechers (recall the discussion on monotonicity), or iii) has lower mean seeding time and, thus, more leechers and less seeders (again by monotonicity). Thus, IoP insertion has a larger impact on swarms with lower total available upload capacity, or equivalently, higher capacity needs.

Results regarding the average download times are presented in Fig. 3; note that we refer to swarms A and B as sA and sB. For scenario A, it can be seen that peers of both swarms experience large performance improvements when the IoP is inserted in swarm B, which serves the large content file. Although counter-intuitive, peers of the other swarm A experience a large performance improvement too. This is probably due to the capacity surplus from multi-swarm peers. That is, peers of swarm B are now served also by the IoP, resulting in less inter-action with multi-swarm peers; thus a

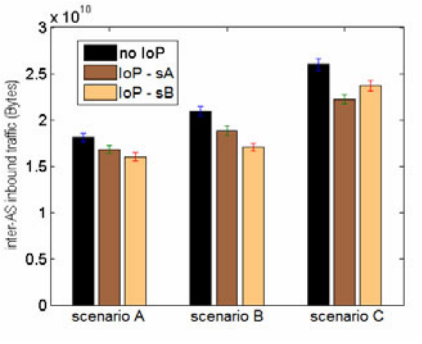


Fig. 2. Incoming inter-AS traffic for AS1

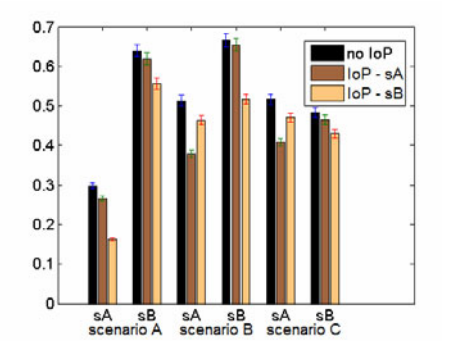


Fig. 3. Average download times for AS1

higher capacity of multi-swarm peers is now available to peers of swarm A. For scenarios B and C, we observe that the IoP achieves larger performance improvements for the swarm it joins in all scenarios, although also peers of the other swarm benefit.

In Fig. 4, the outgoing and incoming inter-AS traffic for AS1 is depicted for scenarios D, E and F, where two tunable parameters are modified in each scenario. We observe that the reduction of the incoming traffic is larger when the IoP serves either i) the swarm with the larger file and higher arrival rate of leechers, or ii) the swarm with the smaller file but the lower seeding time, or iii) the swarm with the higher arrival rate and larger seeding time. The outgoing traffic behaves similarly in the various scenarios. Generally, we can conclude that the impact of the IoP insertion on inter-AS traffic is more significant when it joins a swarm with a larger number of leechers. Also, the effect of the IoP is highly dependent on the arrival rate of leechers; the seeding time is somewhat less important, and the content file size even less.

The average download times for scenarios D, E, and F are shown in Fig. 5. We observe larger performance improvements for the peers of the swarm that the IoP joins in each of these three scenarios; nevertheless, the peers of the other swarm also benefit by the IoP. Note that in scenarios D and E, largest improvement is experienced respectively by peers of: i) the swarm with larger file size and higher arrival rates, or ii) the swarm with smaller file size but also smaller seeding time, when the IoP

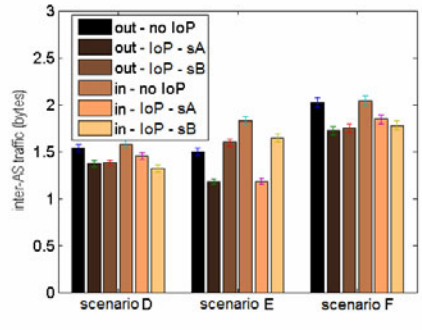


Fig. 4. Out- and incoming traffic for AS1

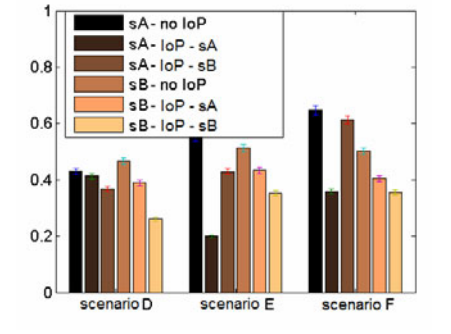


Fig. 5. Average download times for AS1

participates also in them. This happens because the capacity demand is higher in these swarms. Furthermore, scenario F reveals that when the IoP participates in the swarm with the lower seeding time, the percentage of improvement is higher than when it joins the swarm with higher arrival rate. Therefore, we conclude that the effect of IoP on the download times is more affected by the seeding time, somewhat less by the arrival rate and finally even less by the content file size.

4 Bandwidth Allocation

In comparison to the last section, we now discuss the possibility to increase the effectiveness of IoPs in scenarios where its bandwidth has to be distributed on multiple swarms. Thus, we assume that the IoP has already committed itself to a number of swarms using a swarm selection strategy based on our previous results. More specifically, we consider scenarios where the IoP has joined two swarms. This allows us to clearly see which of the swarms benefits the most from the allocated cache capacity. Since here it is our primary goal to understand which bandwidth allocation strategy works best, we choose this simple setup instead of employing more realistic scenarios with many swarms. However, we expect our conclusions to apply to such scenarios too. Finally, we assume that the IoP does not unchoke remote peers, in order to maximize its positive effect on the performance of local peers.

For this evaluation, we introduce different Bandwidth Allocation Algorithms (BWAAs), namely *UNIFORM, MIN, MAX, PROP* and *INV-PROP*. The task of the BWAA is to distribute the IoP's available bandwidth in the specific given scenario. In this context, preferable means the highest possible reduction of inter-AS traffic and download time. Here, *UNIFORM* is the default, straightforward strategy (which distributes the available IoP-bandwidth equally among the swarms), and therefore serves as a benchmark strategy for the new approaches to be compared against.

Except for the *UNIFORM*-algorithm, all distribution concepts are based on a parameter R. This parameter represents the ratio of local leechers to external peers. *MIN and MAX* assign all of the available bandwidth to either the swarm with the smallest or the one with the largest R-value. *PROP* and *INV-PROP* on the contrary use this value to give a proportional amount and respectively an inversely proportional amount of the upload capacity to each of the swarms.

Thus, we cover here a range of different policies, where the two most extreme ones, *MIN* and *MAX*, come close to being Swarm Selection policies, since they in effect let the IoP participate only in a single swarm. In a sense, these two strategies therefore introduce a fourth swarm selection criterion to the three parameters discussed previously, namely R. However, due to the fact that they affect the same mechanism as the other strategies in this section, they are compared to these instead of treated separately. In addition, it is not feasible in practice to allocate the full bandwidth of a cache to one swarm, since a saturation effect may set in that wastes a part of this bandwidth. Therefore, we treat the *MIN* and *MAX* policies as theoretical boundaries regarding the Bandwidth Allocation.

Simulation setup. Our simulations are again based on a two AS scenario, in which the first AS, AS1 represents the point of view of a single ISP and the second AS, AS2, embodies the rest of the world. The analysis involves two swarms that coexist

simultaneously, but do not affect each other (i.e., they have no regulars peers in common), in contrast to the previous experiments. The only connecting instance is embodied by the IoP which enters both swarms. The IoP is equipped with high upload capacity (768 kB/s) which has to be distributed between the two overlays. Compared to the upload capacity of the considered swarms, this capacity is large enough to have an impact on the traffic and the download times of the peers. Still, it does not add too much capacity to the swarm to marginalize the P2P protocol and its effects.

We consider overlay setup similar to the setup described in Section 3 only using meanIAT = 10.0 s. and meanST = 300.0 s; this results in swarms consisting roughly of 120 peers in the steady state. The only difference here is the peer distribution across the two ASs. Again overlay values are common according to recent measurements [12].

In every simulation, our design uses different peer distributions per swarm. Swarm A distributes its peers equally across the two ASs. Swarm B varies the amount which is assigned to the system of the ISP. Therefore, either 5%, 10%, 15% or 30% of the peers in swarm B are located in AS1, which represents the focus of our interest.

In the remainder of this section, we first take one of the possible peer distributions for swarm B and analyze this setup in detail. Subsequently, we take a look at the impact of varying this peer distribution. All results are presented with their according 95% confidence intervals obtained from 10 runs.

Evaluation results for Bandwidth Allocation strategies. First, we focus our discussion on the scenario where AS1 contains 50% of the swarm A peers (as always) and 5% of the swarm B peers, before taking a look at the general picture. In this case, the IoP has the opportunity to distribute its available capacity on two swarms, where one has about 60 local peers and the other has only 10% of that size.

The impact of the BWAAs onto the inter-AS traffic can be seen in Fig. 6. On the x-axis, the four scenarios are displayed, representing the amount of peers that is entering AS1. The graph illustrates the total amount of incoming traffic for AS1.The outgoing traffic is neglected here, as in Section 3, but shows a similar behavior. Fig. 6 also differentiates among the utilized BWAAs. They are emphasized by the use of a color scale where *black* represents the only setup where no IoP was used.

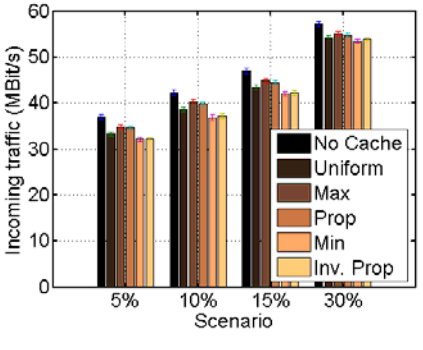


Fig. 6. Inbound traffic for AS1

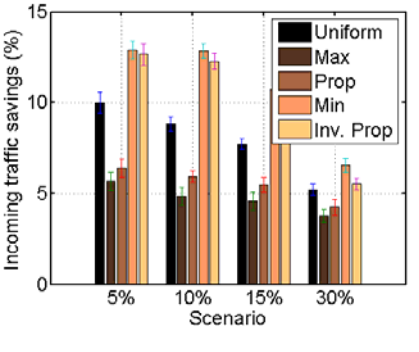


Fig. 7. Incoming traffic savings for AS1

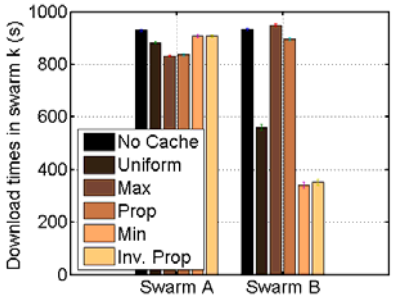


Fig. 8. Download times per swarm for the 5% scenario

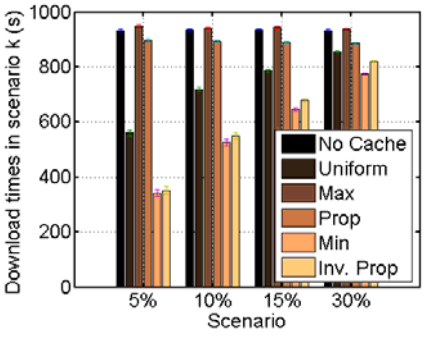


Fig. 9. Download times per scenario

We observe that the introduction of an IoP always reduces the amount of inter-AS traffic regardless of which BWAA was chosen. This effect is not surprising, since we induce additional local upload capacity into the system and particularly in AS1. Among the different algorithms, however, we also note that the amount of inter-AS-traffic is varying. Since the main goal of this section is to optimize the bandwidth-distribution of an existing IoP, we focus our evaluation on the differences between the algorithms. Therefore, we take all results and compare them with the scenario where no IoP is used. In this way, we get an insight into the possible traffic-savings and can easily compare the efficiency of the BWAAs.

This kind of evaluation leads to results as they are illustrated in Fig. 7. The graph shows the results of the incoming inter-AS traffic savings in AS1. The first observation we make is that the efficiency of a IoP is highly dependent on the chosen BWAA. This can be seen in the incoming direction, where the *MAX*-algorithm only saves about 5% of the incoming inter-AS traffic, but this saving can be more than doubled by using the *MIN*- or *INV-PROP*-algorithm. We also note that the savings of the *MAX*-algorithm are even worse than those of a standard uniform distribution. In this (5%) scenario, we achieved the best results by utilizing the *MIN*-algorithm.

This finding implies that an IoP is more efficient in smaller ASs, for the following reasons. If the IoP utilizes the *MAX*-algorithm, it assigns its whole bandwidth to the swarm with the highest ratio of local leechers to external peers. Since 50% of the peers in swarm A are located in the AS 1 against only 5% of swarm B, swarm A will always have a higher number of local leechers, as well as a lower number of external peers. Therefore, it is preferable for the IoP to assign the whole available upload capacity to swarm A. Here, the bandwidth has to be divided between much more peers, than it would have to in swarm B. Subsequently, each one of the peers in swarm A receives only a small share of bandwidth. This leads to a longer download-time and finally to increased inter-AS traffic since the peer spends more time downloading from external sources.

This observation is confirmed as we analyze the results for the peers' download times, which are illustrated in Fig. 8. The graph depicts the mean download times for

each of the two swarms. As we expected, the *MAX*-algorithm leads to a reduction of the download time in the swarm A. However, this decrease is small if we compare it to the possible reductions that can be achieved in swarm B. *MAX* is able to reduce the mean download time in swarm A by about 100 seconds. In contrast, we are able to save over half of the download time (500 seconds) in swarm B, if we use the *MIN*-algorithm. The *UNIFORM* distribution is able to reduce the download times in both swarms but again, we observe that, due to the lower number of peers, the bandwidth is used more efficiently in swarm B. However, it should be noted that in any case, the number of peers experiencing the shorter download times is less in swarm B, since there are less leechers of this swarm in AS1. Combining the results for the traffic performance and the download times, we conclude that in this (5%) scenario, the IoP achieves the highest effectiveness if it uses the *MIN*- or the *INV-PROP*-algorithm.

Next, we take a look at the whole range of our considered scenarios, comparing the results for multiple configurations. As we can see in Fig. 6 and 7 changing the peer distribution of swarm B on the one hand has a large impact on the total inter-AS traffic. On the other hand it also affects the efficiency of each of the BWAAs.

The first observation can be made by analyzing the development of the total traffic exchange over the different scenarios, which is illustrated in Fig. 6. We observe that the total amount of incoming traffic is increasing, as the proportion of peers in the AS1 rises from 5% to 30%. This finding can be traced back to the fact that we allocate a larger portion of swarm B into AS1. This variation leads to more peers in AS1 that have the possibility to up- and download data externally. Therefore, the total amount of inter-AS traffic is rising.

However, the four peer distribution scenarios also affect the individual BWAA efficiency, yet showing the same effect we could already observe for the 5% scenario. Fig. 7 illustrates inter-AS traffic savings compared to the respective scenario, where no IoP is used. In this graph, we observe that inter-AS savings are dropping as the amount of peers in AS1 increases. This finding confirms the assumption of a decrease of the IoP's efficiency in larger ASs. As we compare the values of the first and last scenarios, we note that the savings of the *MIN*-algorithm, which prefers swarm B, are almost cut in half if the amount of local peers in this swarm is increased to 30%.

We also denote a decreased saving regarding the algorithms that prefer swarm A (*MAX* and *PROP*). This dropping appears surprisingly, as we do not modify the swarm affected by these algorithms. However, this observation can be explained by the raised amount of total inter-AS traffic. In every scenario, the algorithms save the same absolute amount of exchanged traffic. Since, however, the total traffic is now increased, the proportional saving is lowered. Thus, for all scenarios, our evaluation shows that the highest reduction of inter-AS traffic can be achieved by the *MIN*-algorithm although the possible savings decrease, as the numbers of local peers in the two swarms converge.

If we now take a look at the download times for all setups, the statistics support our previous conclusions for the 5%-scenario. Fig. 9 shows the download time statistics for the four different scenarios. As we raise the proportion of peers located in AS1, we observe increasing values for those algorithms, which mainly affect swarm B. In the 5%-scenario, the *MIN*-algorithm was able to significantly reduce the download time by about 700 sec, whereas in the 30%-scenario, this reduction is lowered to about 200 sec. If we compare the algorithms among each other, we can conclude that

again, the *MIN*-algorithm delivers the best performance. The *INV-PROP*-algorithm.is also a good choice, and may be preferable if more than two swarms are in place.

5 Related Work

The problem of suboptimal behavior for both ISPs and overlay due to randomized overlay decisions and classic traffic engineering, respectively, has been addressed in other research works in the past. There is extensive literature of approaches that study similar scenarios and have same objectives, namely the reduction of inter-connection costs, in terms of inter-domain traffic, and users' performance improvement. In this section, we discuss some of the most representative ones.

In [13], a centralized entity called the Oracle is introduced. The Oracle provides a service to help peers choose local neighbours. When a peer has several possible sources to download content from, it queries the Oracle, sending a list of IP addresses. The Oracle ranks the received addresses according to ISP metrics, taking locality information into account, and returns the list back to the application.

Another centralized approach (namely, P4P) is proposed in [14]. P4P stands for "Proactive network Provider Participation for Applications". P4P's main objective is to allow cooperation between ISPs and overlay providers in order to boost content distribution and optimize utilization of ISP network resources. The P4P framework consists of an ISP-controlled entity, called iTracker, which provides locality information to overlay application trackers (appTrackers). The information is used by the appTrackers to perform a biased initial peer selection before replying to a request.

An approach of different flavor, called Ono has been proposed in [15]. Ono is a fully distributed solution that exploits existing network information to boost self-organising of the overlay network. It has been implemented as a plug-in for the Azureus BitTorrent client and works by executing periodical DNS lookups to CDN names in order to build and maintain proximity maps. A peer compares its proximity maps to other peer's and calculates the overlap. The Ono approach is purely overlay-based, i.e. no cooperation from the ISP or CDN is required.

The aforementioned non-transparent approaches share similar key objectives to our work. However, regarding their function, they propose either alternative neighbour selection criteria, based on some proximity metric provided either by the ISP, as in [13] and [14], or by a public infrastructure such as CDNs [15]. The IoP, on the other hand, does not make use of any network information; it transparently participates in the overlay, offers "for free" additional resources to other peers, and exploits a native self-organizing incentive mechanism.

A solution that is more closely related to the IoP is proposed in [16]. An ISP offers additional *free* resources to selected users that act in a most underlay-friendly way and thus become able to bias localization of the overlay traffic. The purpose of this action is twofold. First, the operator chooses those peers who upload mostly to local peers, thus, increasing the amount of bandwidth available to local peers. Hence, less data is to be downloaded from distant domains. Moreover, this approach offers an incentive for the peers to get in line with the ISP and behave socially and locality-aware. This solution has several similarities to the IoP, such as the extra capacity made available by the ISP to the network and the fact that it also leads to win-win, since it improves the users' performance too.

6 Conclusions

The insertion of ISP-owned Peers was previously known to be a promising solution for the ISP to avoid the increasing inter-domain traffic charges due to sub-optimally routed overlay traffic and the performance deterioration of peer-to-peer users due to other ISP traffic management practices. In this paper, we investigated by means of simulations the impact of different overlay factors that may affect IoP's effectiveness, and studied policies for the efficient allocation of the IoP's abundant resources to the swarms that it joins. Our main findings are that the inter-arrival rate of peers and their seeding times are more influential inputs of a Swarm Selection policy, w.r.t. the inter-domain traffic and the download time of the peers, than the size of the shared file. As for the Bandwidth Allocation policy employed once the IoP has joined a set of swarms, our results suggest that it is useful to utilize more of the IoP's upload capacity in swarms with a low share of local leechers, since this leads to higher savings of inter-domain traffic. Future work should focus on combining all IoP influential factors in a single index, which is better suited to the case of many swarms.

Acknowledgments. This work has been accomplished in the framework of the EU ICT Project SmoothIT (FP7-2007-ICT-216259). The authors would like to thank all SmoothIT partners for useful discussions on the subject of the paper, as well as the anonymous referees for their constructive comments.

References

1. Liu, Y., Zhang, H., Gong, W., Towsley, D.: On the Interaction Between Overlay and Underlay Routing. In: Proc. IEEE INFOCOM 2005 (2005)
2. The SmoothIT Project, http://www.smoothit.org
3. Papafili, I., Soursos, S., Stamoulis, G.D.: Improvement of bitTorrent performance and inter-domain traffic by inserting ISP-owned peers. In: Reichl, P., Stiller, B., Tuffin, B. (eds.) ICQT 2009. LNCS, vol. 5539, pp. 97–108. Springer, Heidelberg (2009)
4. Cohen, B.: Incentives build robustness in BitTorrent. In: Proceedings of the First Workshop on the Economics of Peer-to-Peer Systems, Berkeley, CA, USA (June 2003)
5. SmoothITSimulator v3.0, http://protopeer.epfl.ch/wiki/BitTorrent
6. ProtoPeer, http://protopeer.epfl.ch/index.html
7. Oversi Networks Ltd. Company website, http://www.oversi.com/
8. PeerApp Company website, http://www.peerapp.com/Default.aspx
9. The SmoothIT project: Deliverable 2.3 – ETM Models and Components and Theoretical Foundations (Final) (October 2009)
10. Lehrieder, F., Dán, G., Hoßfeld, T., Oechsner, S., Singeorzan, V.: The Impact of Caching on BitTorrent-like Peer-to-peer Systems. In: IEEE International Conference on Peer-to-Peer Computing P2P 2010. Delft, The Netherlands (August 2010)
11. Qiu, D., Srikant, R.: Modeling and Performance Analysis of BitTorrent-like peer-to-peer networks. In: Proc. ACM SIGCOMM Conference on Applications (2004)
12. Hoßfeld, T., Hock, D., Oechsner, S., Lehrieder, F., Despotovic, Z., Kellerer, W., Michel, M.: Measurement of BitTorrent Swarms and their AS Topologies. Computer Networks (2011) (in press)

13. Aggarwal, V., Feldmann, A., Scheideler, C.: Can ISPs and P2P users cooperate for improved performance? SIGCOMM Computer Communication Revue 37(3) (2007)
14. Xie, H., Yang, Y.R., Krishnamurthy, A., Liu, Y., Silberschatz, A.: P4P: Provider Portal for Applications. In: ACM SIGCOMM, Seattle, WA, USA (August 2008)
15. Choffnes, D., Bustamante, F.: Taming the Torrent: A practical approach to reducing cross-ISP traffic in P2P systems. In: ACM SIGCOMM, Seattle, WA, USA (August 2008)
16. Pussep, K., Kuleshov, S., Gross, C., Soursos, S.: An incentive-based approach to traffic management for peer-to-peer overlays. In: Stiller, B., Hoßfeld, T., Stamoulis, G.D. (eds.) ETM 2010. LNCS, vol. 6236, pp. 2–13. Springer, Heidelberg (2010)

Efficient Adaptation of Modulation and Coding Schemes in High Quality Home Networks

Hendrik Koetz and Ruediger Kays

Communication Technology Institute, TU Dortmund University, 44221 Dortmund
{hendrik.koetz,ruediger.kays}@tu-dortmund.de
http://www.kt.e-technik.tu-dortmund.de

Abstract. The increasing number of wireless devices exchanging high quality digital multimedia content within the home necessitates high rate, reliable transmission technologies, like IEEE 802.11n. Though providing very high throughput, IEEE 802.11n is not perfectly adapted to the requirements of a wireless multimedia transmission. However, the IEEE 802.11n amendment allows for efficient adaptation of modulation and coding schemes (MCS) to optimize achievable throughput and to fulfill quality of service requirements without any effort for the user. This paper describes an MCS adaptation scheme using information from both PHY and MAC layer with the objective to choose the optimal MCS in an autonomous, fast and robust way which can be applied to both a single link and an entire network. In contrast to existing adaptation schemes, it accounts for volatile channel conditions as well as increased collision probability in dense wireless home networks.

Keywords: IEEE 802.11n, link adaptation, modulation and coding scheme, system organization.

1 Introduction

The number of devices exchanging digital multimedia content within the home has been constantly growing since the emerging of digital media like MPEG audio layer III (MP3), digital versatile Disc (DVD), digital video broadcasting (DVB) or, in general, the increasing usage of real-time entertainment in terms of streaming applications. For the time being, the desire for digital media to be available anytime and anywhere within the home was fulfilled by Ethernet. Since an Ethernet setup is complex and necessitates extensive infrastructural modification, wireless LAN became a cheap and convenient installation alternative which allows for favored portability at the same time. Alongside the increasing quality of digital media, high-performance broadband access technologies to the home like digital subscriber line (DSL) or fiber to the home (FTTH) necessitate reliable, high rate wireless media distribution within the home. Since power line communication (PLC) is installation free but not being able to provide mobility, considerable wireless transmission technologies remain IEEE 802.11, WiMedia ECMA-368, 60 GHz systems or wireless optics. In this paper, we focus on IEEE 802.11n [1] systems by reason of its high market penetration and

C. Bettstetter and C. Gershenson (Eds.): IWSOS 2011, LNCS 6557, pp. 81–91, 2011.

its concluded standardization. IEEE 802.11n's maximum throughput is hardly achievable and is still not sufficiently adapted to the requirements of a high quality wireless multimedia transmission suffering from volatile channel conditions and high node density. Thus, off-the-shelf wireless devices available at the market do not achieve the desired quality of service. Though, IEEE 802.11n offers a powerful toolbox with numerous selectable parameters to satisfy multimedia transmission requirements and to overcome the inherent drawbacks of the wireless transmission channel, namely multipath propagation, fading or interference with other wireless devices. In addition, wireless transmission performance significantly degrades while people crossing the Fresnel zone between antennas.

Despite sophisticated source encoding algorithms, video transmission still makes highest demands on wireless network technology. One noticeable failure per hour is to be considered as just acceptable by the user. A continuous high throughput is crucial for a high quality video transmission as a receiver buffer can compensate for throughput drops only to a certain extent. Amongst others, known adaptation strategies to optimize the quality of a wireless transmission are dynamic frequency selection (DFS) and transmit power control (TPC) [2], a modified MAC (MMAC) [3], frame aggregation or frame fragmentation [4]. In this paper, we are concerned with the dynamic adaptation of modulation and coding schemes to current channel conditions of an ongoing wireless transmission.

IEEE 802.11n offers a total of 153 modulation and coding schemes (MCS) which yield gross data rates from 6 to 600 Mb/s depending on the number of spatial streams (one to four), the size of the guard interval (400 or 800 ns) and the available bandwidth (20 or 40 MHz). In HT format, the OFDM based system uses 56 subcarriers for 20 MHz operation or 114 subcarriers for 40 MHz operation that are modulated using BPSK, QPSK, 16–QAM, or 64–QAM, either with equal modulation (EQM) or unequal modulation (UEQM). Forward error correction coding (convolutional coding) is used with a coding rate of $1/2$, $2/3$, $3/4$, or $5/6$.

Choosing an optimal MCS for a wireless transmission over time implies considering the current channel condition, i.e., the signal to noise ratio (SNR) of the channel and the increased collision probability related to the high density of wireless network devices. Suboptimal choices will either result in an increased packet error rate (PER) or in extended channel busy time. Either way, the overall throughput suffers and the transmission task can only be fulfilled efficiently if transmission parameters are continuously adapted to ever-changing conditions. Applying the proposed adaptation scheme not only improves a single link transmission efficiency but also the efficiency of the whole network, given that the adaptation scheme is applied to each wireless device within the network. An optimal MCS selection adapted to current SNR and high node density leads to a significant reduction of retransmissions and thus, to a reduction of channel busy rate. In other words, neighboring stations less interfere with each other which leads to an increased throughput. This effect will additionally be enhanced by transmit power control and thus, data exchanging devices forming non-interfering subnetworks.

2 Related Work

Numerous approaches adapting the MCS in a wireless IEEE 802.11 network can be found in literature. The approaches can be classified into two categories in principle: those based on channel characteristics [5,6,7], i.e., evaluating the SNR of a channel, and those based on link statistics [10,11,12,13,8,9], i.e., determining the PER of an ongoing link. More precisely, [6,7] propose a two-stage approach as well, but omit any control mechanisms and lack adaptation capabilities to inaccurate parameter settings. Furthermore, [8,9] introduce additional overhead via RTS/CTS exchange or probing to reduce link and network efficiency. However, both categories exhibit at least one main drawback. Evaluating the SNR of a channel is a delicate issue as essential assumptions do not apply. Even if the channel might be considered as reciprocal, the applied hardware is not. Measuring the received signal strength indicator (RSSI) of a received acknowledgment or using dedicated measurement procedures with the objective to estimate the channel quality can therefore be regarded as doubtful or unreliable, respectively. However, apart from lacking reliability, the fundamental idea allows for fast adaptation within one frame duration.

Furthermore, adapting the MCS based on determining statistics of an ongoing link demands for a certain amount of information, i.e., keeping track of PER over certain time. Though, gathering link statistics is highly reliable, the expenditure of time is significantly high as a sufficient number of frames is required for a meaningful evaluation. Furthermore, almost all adaptation scheme do not account for collisions as reason for packet loss. Hence, reducing the MCS leads to increased channel occupancy and thus increased collision probability. For these reasons, single-stage approaches lack either reliability or adaptation speed. The most commonly implemented adaptation scheme, auto rate fallback (ARF) [10], basically decides on the success of a frame transmission. In case of ten consecutive successful frame transmissions, the MCS index is increased, whereas the MCS index is decreased in case of a transmission failure.

3 Simulation Setup

The open wireless network simulator (openWNS) [14] comprises a dynamic event driven system level simulation platform with a modular design and, amongst others, includes an implementation of the IEEE 802.11n amendment. In order to simulate our vision of a wireless home network, we added a channel module which is based on extensive measurements of the dynamic behavior of wireless channels in a real home environment [15]. In contrast, most of existing publications on channel modeling cover only factory or office environments whose characteristics significantly differ from home environments. Our measurements were carried out at 2.4 GHz (channel numbers 1, 5, 9, and 13) over 24 h, including line of sight (LOS) and none line of sight (NLOS) transmission. To begin with, we determined static attenuation for different links, for example from living room to basement or from workroom to bedroom. Then, we determined

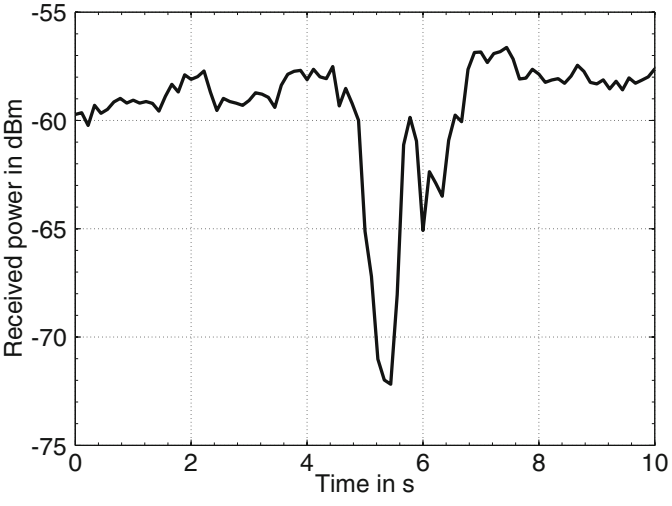


Fig. 1. Fluctuation of received signal power over time in wireless communications

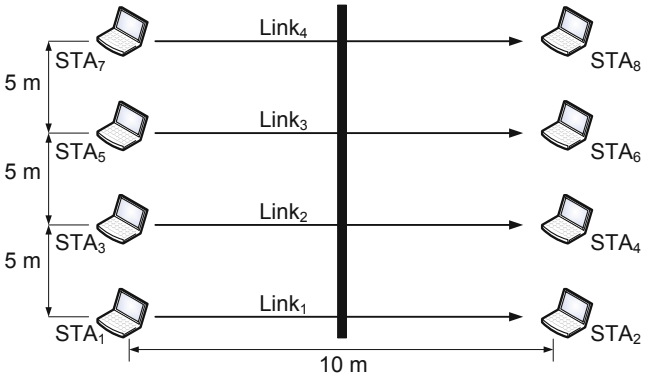


Fig. 2. Scenario setup

dynamic attenuation due to temporal variations of received power. A meaningful extract of our measurements for one link is depicted in Fig. 1 and comprises several deep fades up to 16 dB and exemplifies a volatile wireless transmission in a dense home network. Each link within the simulated wireless home network experiences the same dynamic channel fading, slightly deferred in time, with different static attenuation according to our measurements. The parameters of our channel measurements match the IEEE 802.11n channel model B [16, 17] for residential homes and small offices, using one cluster, no Doppler spread for LOS and one cluster, no Doppler spread, Rician factor of -3 dB for NLOS transmission.

A realistic wireless home network comprises several wireless devices such as media servers and media renderers which are densely distributed within a typical

distance of about 5 to 10 m. We do not expect a central access point to be available for network organization. Although a wireless access point is most likely to be available at almost any home network, we expect an access point to provide internet access only. In order to maximize link and network efficiency, we assume high quality transmissions primarily in ad-hoc mode without any central coordination. Hence, our simulation setup comprises eight stations, i.e., four concurrent links using single spacial streams (MCS 0 to 7) with a fixed transmit power of 20 dBm, a guard interval of 800 ns and a channel bandwidth of 20 MHz. The payload length amounts to 1500 byte and enhanced distributed channel access (EDCA) was used as channel access mechanism with access category video (AC_VI). The complete simulation setup is depicted in Fig. 2.

4 Two-Stage MCS Adaptation Scheme: nDIRA

Our approach to a fast, robust and self-organizing dynamic MCS adaption scheme without any effort for the user comprises a two-stage design as depicted in Fig. 3, named nDIRA: 11n dual instance rate adaptation. The targeted adaptation speed is provided by a first stage: the PHY decider. A second stage, the MAC decider, ensures reliability and robustness. The adaptation process takes place at transmitter side using most suitable information from both MAC and PHY layer. The optimal MCS can be found within one frame duration and the selection process is unnoticeable to the user.

4.1 PHY Decider

At the beginning of the decision process, the channel quality, i.e., the current SNR of the channel, has to be estimated by means of determining the received signal strength indicator (RSSI) of a received acknowledgment. At fixed transmission power, the RSSI of a received acknowledgment indicates the path loss between sender and receiver. Although IEEE 802.11e/k/n provide several more precise feedback metrics for channel estimation, we do not consider any since they introduce additional overhead, and thus reduce transmission efficiency considerably. Especially in high quality home networks we are targeting, any additional overhead has to be avoided as possible. Once channel quality has been determined in terms of SNR, the PHY decider selects the most efficient MCS by means of a threshold value decision. As we are targeting a high quality and efficient wireless transmission, we previously developed an efficiency measure η_{L} [18] which provides the essential thresholds for the threshold value decision. Basically, the efficiency measure η_{L} is defined as:

$$\eta_{\mathrm{L}} = \frac{1}{N_F} \cdot \sum_{i=1}^{N_F} \frac{PL_i}{t_{f,i} \cdot B \cdot \log_2\left(1 + \frac{P_{T,i} \cdot a}{(B \cdot N_0 + P_{Int}) \cdot I}\right)} \tag{1}$$

In general, this efficiency measure η_{L} is the ratio of effort to result of a wireless transmission and is based on the Shannon-Hartley theorem which established

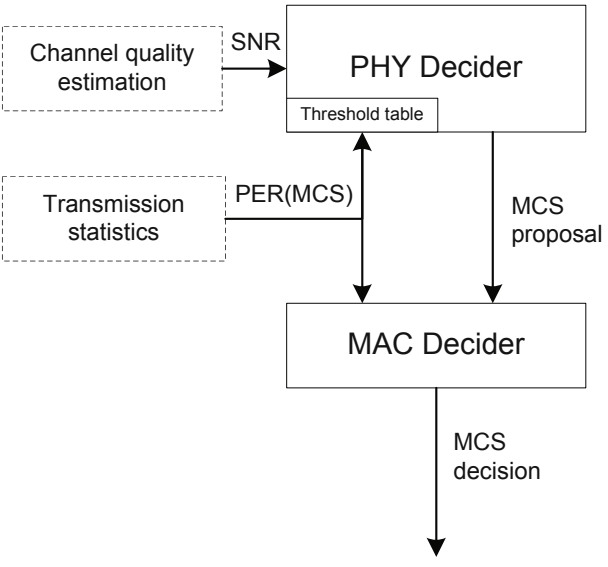


Fig. 3. Two-stage design approach for efficient link adaptation: nDIRA

Shannon's channel capacity [19] for a communication link. The efficiency measure η_{L} considers the number of transmitted payload bits PL, duration of channel occupancy t_F, allocated bandwidth B, transmit power P_T, transmission factor a, spectral density of noise N_0, interference level during frame transmission P_{Int}, implementation margin I and the number of transmitted frames N_F. The index i indicates the i-th frame. Fig. 4 shows an evaluation of the efficiency measure η_{L} over SNR for MCS 0 to 7. Each maximum represents the most efficient working point, and hence the PHY decider's thresholds. This evaluation has been carried out using simulation parameters as described in Section 3. As can be seen from (1), η_{L} is dependent on the simulation parameters. Therefore, numerous threshold tables exist which are in each case selected according to the current scenario. The efficiency measure η_{L} is used in the forefront only for threshold definition and will not be recalculated while ongoing transmissions. For threshold definition, the fraction $\frac{P_{T,i} \cdot a}{(B \cdot N_0 + P_{Int}) \cdot I}$ represents the SNR within our simulations.

According to our experience with AWGN channels, the thresholds based on channel model B appear to be very demanding, i.e., the thresholds are very high for corresponding MCS. Reason for this is the challenging channel model B which originates poor bit error rates even at high signal to noise ratios. Thresholds based on AWGN channel are about 50 to 70 % lower than the thresholds based on channel model B.

4.2 MAC Decider

The decision of the PHY decider on the MCS of the upcoming frame is then forwarded to the MAC decider as a proposal, still being aware of its inaccuracy.

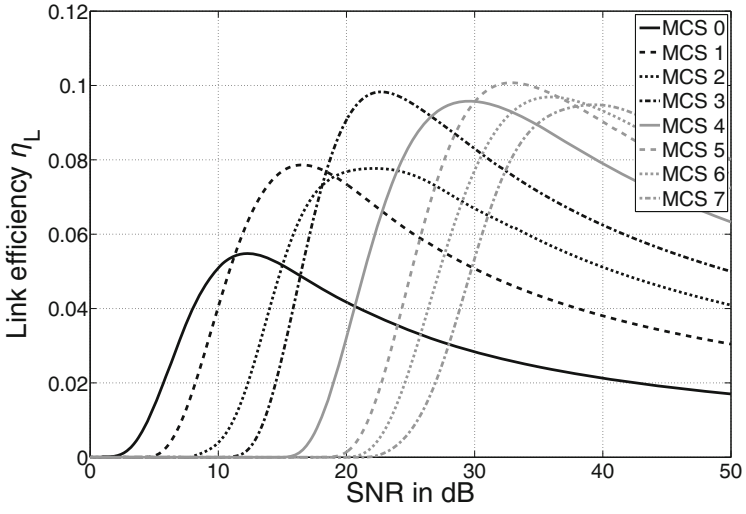


Fig. 4. Efficiency measure η_L over SNR for MCS 0 to 7 at 20 MHz bandwidth, 800 ns guard interval, 1500 byte payload length and channel model B

Hence, a second stage, the MAC decider is necessary to either verify the MCS proposal or overrule it, when appropriate. A feasible method to obtain reliable link statistics is to track transmission parameters like PER or number of received and missing acknowledgments. In contrast to the channel quality estimation, these parameters are highly reliable. In order to analyze the obtained link statistics, the MAC decider makes use of an exponential moving average (EMA) [20] of past transmission statistics. An EMA comprises an infinite impulse response filter which applies exponentially decreasing weighting factors λ^n to transmission parameters x_{t-n} and is given by:

$$\text{EMA} = \frac{x_{t-1} + \lambda \cdot x_{t-2} + \lambda^2 \cdot x_{t-3} + \ldots + \lambda^{n-1} \cdot x_{t-n}}{1 + \lambda + \lambda^2 + \ldots + \lambda^{n-1}}. \tag{2}$$

Due to the time variant wireless channel, current transmission statistics are more meaningful than statistics from the past for channel quality estimation. In order to account for this fact, the weighting factor λ can be used to adjust the weighting degree. According to our simulations, a $\lambda = 0.75$ and n = 100, i.e., the window size, turned out to be reasonable. An overruling becomes necessary in case inefficient MCS have been repeatedly chosen in the past, thus inducing an increased PER. This case will be detected by the MAC decider by constantly determining the EMA of each MCS.

Once EMA of an MCS exceeds a sensitivity level ϵ, the proposed MCS will be considered inefficient. Therefore, ϵ can be used to adjust the MAC decider sensitivity, a value of $\epsilon = 0.5$ proved to be convenient. Several alternatives are conceivable to select the *next best* MCS. We propose to select the next lower MCS index because it promises to be more robust at slightly decreasing data rate. To

avoid such overruling in the future, we suggest to simultaneously increase the corresponding threshold that has selected the inappropriate MCS in the first place. An increment of 0.5 to 1 dB is to be considered as appropriate, depending on the RSSI resolution of the applied hardware.

In first simulations with perfectly adapted parameters and thresholds, virtually no overruling occurred. In order to determine an appropriate sensitivity level ϵ for hardware impairments or inaccurate parameter settings, we provoked a system reaction by intentionally lowering the thresholds. At the beginning of such an erroneous transmission, numerous overruling occurred. Certainly, due to the self-acting threshold calibration, the system adapts to the applied scenario and the number of overruling significantly diminished to practically zero. Hence, the MAC decider is the final instance in the decision process and provides reliability and robustness by verifying the PHY decider's proposed MCS and by updating imperfect thresholds.

5 Simulation Results

In order to demonstrate the performance gain of the proposed adaption scheme, ARF was used as basis for comparison because it is widely-used in current hardware implementations. The reason for its popularity is its easy implementation despite all known performance drawbacks. Fig. 5 exemplarily depicts a performance comparison of ARF versus nDIRA in an 1x1 SISO setup using single spacial streams MCS 0 to 7. Furthermore, the simulation setting of this network is as described in Section 3, i.e., four concurrent and thus interfering links without central AP. Simulation results are valid only for the described scenario and may vary if other simulation parameters and scenarios are applied.

Note, the aggregated throughput of all four links is outlined over the offered load of a single link within the described scenario. The offered load is varied, starting at zero, searching for the saturation point. The saturation point denotes the point from which on the offered load cannot be completely processed and hence, the throughput flattens or decreases. The saturation point can be efficiently found using the binary search option of openWNS. Instead of selecting different offered loads by hand, binary search starts with an initial value which is doubled in ongoing simulations. This continues until an upper and lower bound are found. The binary search algorithm aborts in case the aggregated throughput does not increase anymore. Fig. 5 reveals that ARF performs comparably in undemanding scenarios below 3 Mb/s offered load per link. Even in more demanding scenarios of up to 10 Mb/s offered load, ARF performs marginally worse. However, in demanding scenarios with offered load exceeding 10 Mb/s per link, ARF is clearly outperformed by nDIRA.

In case of ARF, the saturation point is reached roughly at 9.5 Mb/s. Beyond this point, the aggregated throughput flattens and rapidly decreases thereafter due to the high number of collisions and suboptimal MCS selection behavior. In contrast, nDIRA reaches its saturation point at 12.5 Mb/s, but is still able to offer am aggregated throughput of about 35 to 40 Mb/s afterwards. Thus,

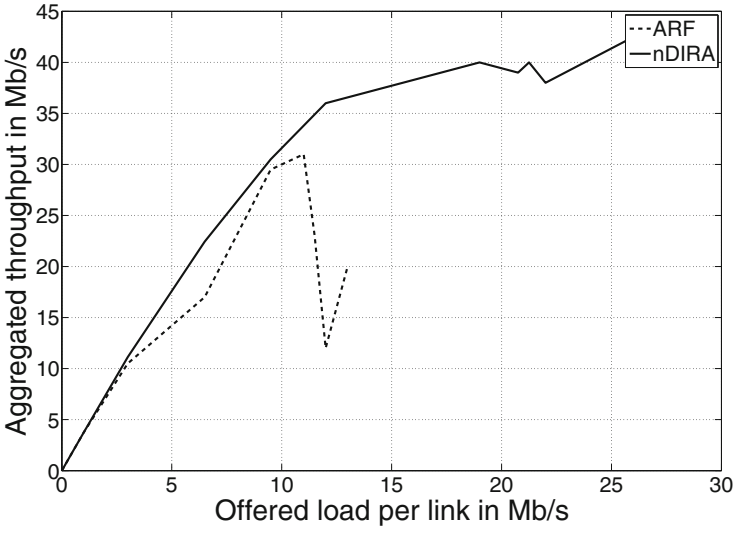


Fig. 5. Performance comparison ARF - nDIRA

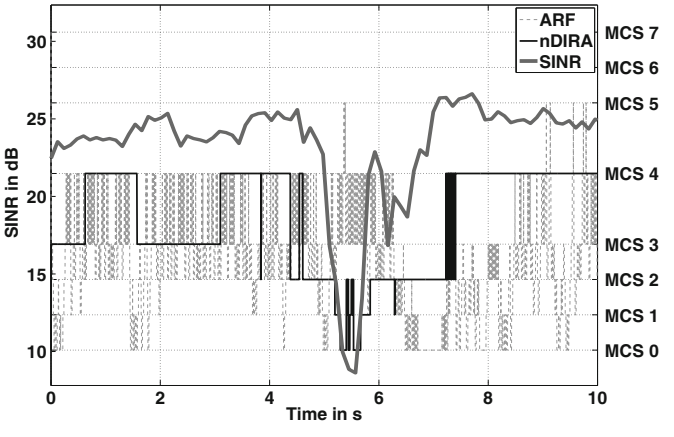


Fig. 6. Comparison of MCS selection behavior

nDIRA is capable of handling a much higher offered load and is less affected by the increased collision probability or volatile channel conditions, unlike ARF. It becomes obvious that a single-stage adaptation scheme like ARF is not sufficiently capable of providing high quality media transmission in dense wireless networks with high collision probability. In contrast, our proposed adaptation scheme nDIRA allows for fast adaptation speed and robustness at the same time. These favored characteristics lead to an increased throughput of the whole network which is capable of accommodate high quality media streams within the home.

An exemplary comparison of the MCS selection behavior for ARF and nDIRA for a single link is depicted in Fig. 6. As can be seen from the figure, ARF shows a strong fluctuating selection behavior, whereas nDIRA operates in a stable way, i.e., avoiding inappropriate MCS transitions. According to our set of simulations, the number of retransmissions can be significantly reduced by approximately 78 % for szenarios exceeding 5 Mb/s offered load per link. It becomes obvious, that a stable selection behavior in combination with a correct or revised channel analysis leads to a significant reduction of packet loss and thus, a decisive performance gain, especially in wirelss networks with volatile channel conditions and high node density.

6 Conclusion

A high quality wireless home network with high node density necessitates efficient transmission technologies, like IEEE 802.11n. Besides, sophisticated adaptation schemes are crucial to deal with volatile channel conditions and high collision probability in order to allow for quality of service without any effort for the user. All known single-stage approaches exhibit at least one main drawback. In particular, the auto rate fallback adaptation scheme revealed that single-stage approaches perform reasonable in undemanding wireless networks but fail in challenging scenarios we have in mind for future high quality home networks. Therefore, we propose a two-stage design approach, applicable to IEEE 802.11n networks, which allows for both fast and robust adaptation of modulation and coding schemes to volatile channel conditions by the use of information provided by both PHY and MAC layer without being affected by an increased collision probability of a high node density. Hence, the proposed adaptation scheme represents a local optimization in terms of a global efficiency optimization of the whole network. Provided that nDIRA is implemented in each device within a wireless home network, the optimization of single links simultaneously entails an efficiency optimization of the whole network. The number of retransmissions can be significantly reduced as well as channel busy time and thus, interference is minimized. This outcome will be enhanced if TPC is applied. Reducing transmit power of nearby communicating devices leads to non-interfering subnetworks, for example suitable for short range communication from a set-top box to TV. In this case, transmit power is reduced to a minimum that is defined by the offered load and the required MCS. In terms of link and network efficiency, the application of a central AP is therefore not beneficial. An efficient adaptation scheme like nDIRA can be the decisive factor for a link fulfilling QoS requirements or for an additional link being allowed to join a wireless network.

References

1. IEEE 802.11n-2009: Enhancements for Higher Throughput (2009)
2. IEEE 802.11h-2003: Spectrum and transmit power management extensions in the 5 GHz band in Europe (2003)

3. Schilling, C., Kays, R.: HOMEPLANE: An architecture for a wireless home area network with management support for high quality of service. In: Conf. Rec. IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications (2008)
4. Hoffmann, O., Kays, R.: Efficiency of Frame Aggregation in Wireless Multimedia Networks Based on IEEE 802.11n. In: International Symposium on Consumer Electronics (2010)
5. Pavon, J.: Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement. In: Proc. IEEE ICC (2003)
6. Haratchev, I.: Automatic IEEE 802.11 Rate Control for Streaming Applications. In: Wireless Communications and Mobile Computing, pp. 421–437 (2005)
7. Xi, W.H., Munro, A., Barton, M.: Link adaptation algorithm for the IEEE 802.11n MIMO system. In: Das, A., Pung, H.K., Lee, F.B.S., Wong, L.W.C. (eds.) NETWORKING 2008. LNCS, vol. 4982, pp. 780–791. Springer, Heidelberg (2008)
8. Wong, H.Y., Yang, H., Lu, S., Bharghavan, V.: Robust Rate Adaptation for 802.11 Wireless Networks ACM Mobicom (2006)
9. Kim, J., Kim, S., Choi, S., Qiao, D.: CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs INFOCOM 2006. In: Proceedings of 25th IEEE International Conference on Computer Communications (2006)
10. Kammerman, A.: WaveLan-II, A high-performance Wireless LAN for the unlicensed band. Bell Labs Technical Journal, 118–133 (1997)
11. Lacage, M.: IEEE 802.11 Rate Adaptation: A Practical Approach. In: International Conference on Modeling, Analysis and Simulation of Wireless Systems (2004)
12. Onoe, A.: Madwifi: Multiband Atheros Driver for WiFi, http://madwifi.sourceforge.net
13. Bicket, J.: Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In: International Conference on Mobile Computing and Networking (2005)
14. Bueltmann, D.: openWNS - open Wireless Network Simulator. In: Electronic Proceedings of 15th European Wireless Conference 2009, pp. 205–210 (2009)
15. Aznar, B.: Dynamic Characteristics of Wireless LAN Channels for Multimedia Home Networks. In: The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC (2007)
16. IEEE 802.11 Wireless LANs: TGn Channel Models. IEEE 802.11-03/940r4 (2004)
17. Schumacher, L., Pedersen, K.I., Mogensen, P.E.: From antenna spacings to theoretical capacities - Guidelines for simulating MIMO systems. In: Proc. PIMRC Conf., vol. 2, pp. 587–592 (2002)
18. Hoffmann, O., Kays, R.: A Link Level Efficiency Measure for Wireless Home Area Networks. In: Proceedings of GLOBECOM (2009)
19. Shannon, C.E.: The Mathematical Theory of Communication. University of Illinois Press, Urbana (1948)
20. Wuensche, O.: Exponentially Weighted Moving Average (EWMA). Eberhard Karlsruhe Universitaet Tuebingen (2005)

Distributed Graph Clustering for Application in Wireless Networks

Chia-Hao Yu[1], Shaomeng Qin[2], Mikko Alava[2], and Olav Tirkkonen[1,3]

[1] Department of Communications and Networking, Aalto University, Finland
[2] Department of Applied Physics, Aalto University, Finland
[3] Nokia Research Center, Helsinki, Finland
{chiahao.yu,shaomeng.qin,mikko.alava,olav.tirkkonen}@tkk.fi

Abstract. We consider distributed clustering of weighted graphs. Each node in the graph is represented by an agent, with agents independent of each other. The target is to maximize the sum weight of intra-cluster edges with cluster size constrained by an upper limit. To avoid getting stuck in not-too-good local optima, we approach this problem by allowing bad decision-making with a small probability that is dependent on the depth of local optima. We evaluate performance in a setting inspired by self-organizing coordination area formation for coordinated transmission in wireless networks. The results show that our distributed clustering algorithm cab perform better than a distributed greedy local search.

Keywords: Distributed graph clustering, Flat clustering, Self-organized network, Non-clusterhead-based clustering, Distributed coordination area formation.

1 Introduction

Graph clustering, which by definition groups vertices of a given input graph into clusters based on underlying structures, has grown popular and found its application in e.g., database systems, biological and sociological networks, and information networks [16]. Broadly speaking, we can divide graph clustering methods into *global clustering* [7,13,14] and *local clustering* [2,15]. A global clustering method assigns each vertex on a given graph to a cluster, whereas a local clustering method only assigns a certain subset of vertices on a given graph to clusters. For application in wireless networks, a global clustering output is usually required. Popular examples include Wireless Sensor Networks (WSNs) and Mobile Ad hoc NETworks (MANETs) [12,3,18] whose network structural properties and requirements demand node clustering for efficient operations.

Due to the enormous complexity and potential risks associated with the operation and management of systems nowadays, self-organization is an increasingly important feature for future mobile networks. It applies to WSNs and MANETs whose network operations are based on ad hoc discovery and dynamic routing between network nodes. As self-organization is crucial to the operations of such networks, node clustering, which can be mapped to graph clustering, should also be approached in a self-organizing manner.

In wireless networks, graphs considered for clustering are weighted graphs with interference couplings which are positive real numbers. Nodes in the graph

C. Bettstetter and C. Gershenson (Eds.): IWSOS 2011, LNCS 6557, pp. 92–103, 2011.

consist of transmitters and receivers who communicate with each other, i.e., a transmitter-receiver pair, and each node corresponds to an individual agent in a self-organizing algorithm. Due to the broadcast nature of radio communication, in principle all receivers are disturbed by all transmitters, and the interference graph is in principle *complete*. The graph is, however spatially embedded–the average weight between two nodes decreases with increasing distance.

In MANETs, network *scalability* problems make hierarchical routing more preferred than flat routing. Hierarchical clustering that directly fits the structure of hierarchical routing has been extensively investigated. In principle, the same argument of dynamic routing applies to WSNs. While clusterheads may not be necessary in MANET, they are usually required in WSNs. In WSNs, network lifetime is one of crucial considerations. A clusterhead-based clustering can be more energy efficient by letting clusterheads take charge of inter-cluster transmission and all cluster members to act as clusterhead in turn.

The introduction of Coordinated Multi-Point (CoMP) transmission in mobile networking [1] has found another application of graph clustering in wireless communications. In CoMP transmission, multiple transmitters coordinate their transmission to either avoid interference between transmissions, or to generate a Multi-Input Multi-Output (MIMO) channel between different cellular transmitters and receivers [10]. Since the number of transmitters involved in CoMP transmission cannot go up unlimited, a Coordination Area (CA), in which coordination between transmitters takes place, is defined. Another reason for having CAs is to reduce the burden of interference measurement for mobile devices. This indicates an upper limit on how many transmitters can cooperate, i.e., the CA size, since the size of a CA cannot be larger than the number of measured interferences. The formation of CA is inherently a clustering problem. Below, we elaborate clusterization problems arising from here.

In the context of Self-Organizing Networks (SONs), clustering methods to our concern should be *online* and *iterative*, so that the cluster assignments are made based on only the knowledge of previously encountered information, and the clustering process is continuous and cluster assignments made now can be changed later. Considering network scalability, *distributed* clustering where each node runs the same algorithm with local information ensures the efficiency of clustering process. In addition, since the interaction between farther away nodes is weak in wireless networks, taking into account only local information should not allow significant loss from global information. We assume our clustering process is run *asynchronously* among nodes, i.e., an algorithm is run by one node at one time, to allow proper reaction of each node to the environment. Nevertheless, since interference situation is changing gradually, the clustering should be done within relatively short time compared to the coherent time of mutual interference. It is noted that there should be a working output from the distributed clustering algorithm for each node to follow, for any given time.

The requirements on clustering methods for CA formation problem suggests a clustering method to cut weak interference coupling is to our flavor. This connects CA formation to minimum-cut problems, and therefore maximum-flow problems with the well-known connection with minimum-cut problems [4,9]. However, due to scalability issue, global clustering is in general not preferred

for a very big graph. Secondly, for minimum-cut algorithms, a source and a sink are required for the algorithms to proceed. This is in general hard in our application. In addition, minimum-cut algorithms use divisive clustering, meaning a graph is sequentially divided into partitions. Due to coordination overhead and growing complexity with respect to involved transmitters for CoMP transmission, a big cluster is not practically favored. This indicates a longer delay before a working clustering can be reached by this approach. Similar argument on divisive clustering methods is applied to the famous Kernighan-Lin bisection algorithm [11]. In addition, Kernighan-Lin methods in general divide nodes into two equal-sized subsets. As we have no prior information on the size of the network, special cares are required if bisection methods are used.

With the properties to be fulfilled above, we consider to reach global clustering via local clustering [2,15]. This can be done by, e.g., initiating the procedure multiple times using each vertex as the seed vertex once, and then combine the local clusters into a global clustering according to some quality measure. While some local search procedures, such as hill-climbing and simulated annealing, are required for local clustering algorithms [2,15] because of big cluster size, this can be avoided in concerned CA formation.

In this paper, we study distributed graph clustering with application to CA formation for CoMP transmission, and we classify it as distributed self-organized clusterization. We assume each vertex can learn local information within its neighborhood. The distributed clustering is negotiation-based so that each node can invite other nodes to join its cluster. By exchanging local information, the inviting and invited nodes can negotiate their local clustering. The clustering does not need to be hierarchical since CoMP transmission in each cluster is independent. We do not consider clusterheads election since it is not compulsory in CoMP transmission. In fact, a clustering output with clusterheads will jeopardize the negotiation process since clusterheads, who usually have dominant right, can decide its action regardless of opposing opinions. This leads to local optima easily because of more limited information for local decisions. We note that clustering algorithms in literature do not allow immediate application to our problem from our previous discussion.

As we have mapped a CA formation problem for CoMP transmission in wireless networks to a node clustering problem, we will refer to CA and cluster as the same concept and use them interchangeably. Although the clustering problem is emerged from the original requirement on CoMP transmission, an evaluation of the system performance requires detailed description on CoMP type as there are different variants for CoMP transmission. Due to the space limit, the proposed distributed graph clustering method will be evaluated using common figure of merit in clustering problems.

The rest of the paper is organized as follows. In Section 2, the distributed graph clustering problem is introduced, its connection to CA formation in wireless networks together with the modeling of cellular networks with weighted graphs are discussed. In Section 3, we present the proposed distributed clustering algorithm. In Section 4, we present results concerning the proposed distributed clustering algorithm on interference graphs. Conclusions and future work are summarized in Section 5.

2 System Model

2.1 Distributed Graph Clustering

The graph clustering problem can be formulated as follows. Let a graph $G = G(V, E)$ of vertices $v \in V$ and edges $e \in E$ connecting two vertices be given. The vertices consist, for example, of collections of wireless network elements, which may be interchangeably called nodes, or cells. We assume all nodes use the same resources so that edge $(v, u) \in E$ for every $v \in V$, $u \in V$, and $v \neq u$ (i.e., our graph is complete). To each edge $(v, u) \in E$ there is a corresponding weight $w(v, u) \in \mathbb{R}_+$ indicating the strength of interference from vertex u to vertex v. The objective in graph clustering is to find a clustering so that we have minimum inter-cluster cut, where the sum weight of edges between different clusters is minimized, and the cluster size is limited to less than a fixed number.

In addition to the formulation above, we want to perform the node clustering in a self-organized and distributed manner so that each cell decides which cluster to go for based on local information (gathered either through passive interference observation or through a negotiation protocol) only. In general, these decisions take place inside a routine, which each node executes independently an identical copy of. It is assumed to have access to all the local information necessary, such as the interference caused by the neighboring nodes together with any data the nodes choose to broadcast to their neighbors. Any information sent by a node to its neighbors is assumed to be immediately available to the neighboring nodes, i.e., no transmission delays are modeled.

A common measure for clustering results involves *density* of inter-cluster and intra-cluster. With weighted graph, the definition of density in [16] involves number of edges, in addition to edge weights of a clustering output. Since our graph is complete, we ignore the number of edges and define a density measure as the ratio of summed intra-cluster weights to total weights by

$$Q = \frac{\sum_{\mathcal{C} \in \mathfrak{C}} \sum_{(v,u) \in E, v \in \mathcal{C}, u \in \mathcal{C}} w(v, u)}{\sum_{(s,t) \in E} w(s, t)}, \tag{1}$$

where $w(v, u)$ is the weight of edge (v, u), $\mathfrak{C}$ is the output of clustering, and $\mathcal{C}$ is a cluster in $\mathfrak{C}$.

The motivation of distributed graph clustering with such settings is for application in CoMP transmission in wireless networks. With precious spectrum resources, one should make efficient use of them spatially (i.e., the same frequency bands should be reused in space as often as possible). This makes mutual interference between co-channel users stronger. Our transmission environment would become interference-limited, rather than noise-limited. In an interference-limited environment, raising transmit power does not increase Signal-to-Interference plus Noise Ratio (SINR), which is a crucial performance metric. The introduction of CoMP transmission is to coordinate the transmissions from different transmitters so that the mutual interference does not jeopardize our transmissions.

Optimally, the best solution for CoMP transmission would need cooperation between all transmitters over a whole communication network. However, the complexity for global cooperation is very high and the expected additional gain

of global cooperation over local cooperation is usually limited. Because of this, CoMP transmission is defined to be within a CA. Transmitters within a same CA shall coordinate their transmissions, whereas transmitters of different CAs shall treat each other as interferers. Our distributed graph clustering is to solve the formation of CAs for CoMP transmission in wireless networks.

For a transmitter to learn the interference situation of its receivers, interferences from different interferers are measured at receivers and then reported to the transmitter through a feedback channel. From practical points of view, interference information fed back to a transmitter should be kept to a reasonable amount. In addition, there is a limit on how many interferers one receiver is capable of measuring. If the receiver is a mobile device, this limit can be set so that the interference measurement task is not too big a burden for the mobile device. This hard limit immediately defines a limit on the cluster size.

2.2 Connection to Wireless Networks

The graphs addressed in this paper can be related to snapshots of networks of mobile radio Transmitter-Receiver elements (TRx). Each TRx consists of a transmitter and its corresponding receivers. While a transmitted signal is sent from transmitter in a TRx, the reception is performed by receivers of the TRx. To describe the interference between different TRxs, a statistical number based on the observation of all related receivers is defined. In this paper, the terms TRx and node are used interchangeably.

In a snapshot, each TRx is located at a specific location x_i. From the transmission by each TRx, all others receive a (differently) attenuated version due to the broadcast nature of the radio channel. Being indistinguishable from thermal noise, weak signals can be neglected. The snapshot of the network is a weighted directional graph with edges starting from a TRx and ending in another with the interference powers between the corresponding transmitter and receiver as the edge weights. Typically, these are modeled as consisting of a deterministic, distance-dependent component, and a random component due to shadowing and fading, e.g.,

$$I_{jk} = P_k \|\mathrm{x}_j - \mathrm{x}_k\|^{-\alpha} \xi_{jk}, \tag{2}$$

where I_{jk} is the interference power from node k to node j, P_k is the transmission power of node k, α is the path loss exponent and ξ_{jk} is the random component of the path loss [17], which is correlated in the spatial domain. Typically $\alpha \geq 2$.

From a snapshot of a mobile network, an interference graph can be constructed by collecting a group of closely coordinated elements to a node. Examples of a node in such an interference graph are given by a Base Station (BS) in a cellular network, together with the mobile stations it is serving, or a WLAN Access Point, together with its clients. The edge weights in such graphs are based on statistical numbers describing the statistics of the interference experienced at receivers at a node from transmitters in other nodes, according to Eq. (1). To measure the relevance of interference experienced, it should be compared to the signal level experienced in communications inside the node. In the cellular and WLAN settings, a natural measure of interference experienced at a mobile station/client, is the ratio of the interference power caused by the BS in another

cell to the signal power of the own BS. Both of these quantities would follow (1). Considering a mobile station m at distance d_{om} from its serving BS o, and at distance d_{nm} from an interfering BS n, a measure of the interference power caused by this neighbor to m would be

$$I_{on,m} = (\frac{d_{om}}{d_{nm}})^{\alpha} \frac{\xi_{nm}}{\xi_{om}}, \tag{3}$$

assuming the same transmit power for the interfering and serving BSs.

If there is only one receiver and transmitter per node, the interference coupling between o and n is directly determined by (3). If there are multiple transmitters and/or receivers, a statistical measure of the corresponding interferences (3) for different m served by o should be used. We shall simply use the maximum experienced interference as an interference coupling so that $I_{on} = \max_{m \in \mathcal{S}_o} I_{on,m}$, where $\mathcal{S}_o$ is the set of mobile stations served by BS o. Usually, the strongest BS is chosen to serve a mobile station, so that $I_{on,m} \leq 1$. An important consequence of this model of interference coupling is that generically, the further away an interferer is, the weaker is its interference.

It is assumed here that the interference couplings have been symmetrized leading to symmetric real-value interference network and therefore a symmetric, i.e., undirected, graph with $I_{on} = I_{no}$. Unless guaranteed by other properties of a system, such symmetrization can be achieved by inter-node signaling.

Up till now, an interference matrix I with entries I_{on} describing the interference price between node n and node m can be constructed. The interference prices are used as the weight of edges and therefore $w(o, n) = I_{on}$ when calculating the density measure defined in (1).

2.3 Weighted Graphs Modeling Cellular Networks

As a study case for distributed clustering, we construct a weighted graph describing a network according to typical system simulation prescriptions used for evaluation of wide area cellular networks [5]. Base stations are placed on vertices of a triangular grid, so that the corresponding cells are hexagonal. We consider a single-slope path-loss model (2) with path loss exponent $\alpha = 3.76$. Shadow fading standard deviation is assumed to be 3 dB, and shadowing correlation between cells is 0.5. Note that shadow fading in a wide area cellular network is to take into account the impact of terrain, obstructions etc. on the path loss. We consider an interference limited network, where the cell radius is small enough to render thermal noise irrelevant. In simulations we have used 1 km cell radius.

We uniformly drop 8 users per cell on average to the coverage area of the cellular system. Cell selection is performed for all mobile stations; the serving cell is selected based on the smallest total path loss including the deterministic distance-dependent path loss and shadow fading. The interference coupling is then based on symmetrized worst couplings, i.e., the coupling is the worst of the relative interferences from transmissions of base station n experienced by the users served by o and from transmissions of base station o experienced by users served by n.

Note that in a wireless communication system, a more straightforward interference measure should be Carrier-to-Interference-plus-Noise Ratio (CINR) on the interference links. In 3, the additive while gaussian noise and the interference from other BSs are ignored. The use of (3) here is for better illustration while still capture the essence of CINR.

3 Neighborhood Negotiation with Monte Carlo Method Clustering

Here, we describe our clustering algorithm denoted by Neighborhood Negotiation with Monte Carlo Method (NNMCM). Each node in our problem runs this algorithm in a fully asynchronous manner, indicating that the time it takes for communicating and negotiating with the neighboring nodes and to decide the local clustering are small compared to the time of iteration, which is the period between the time a node decides on its clustering. Due to our application, we assume an initial clustering where each node belongs to a unique cluster. As clustering routine runs, nodes are merged into same cluster so that the total number of clusters decreases. It is also possible that one node decides to stay in one cluster in current iteration but changes to another cluster in later iteration. In the classification proposed in [16], NNMCM is a localized version of agglomerative clustering.

In real-value weighted graph, it is unlikely that we run into a plateau during clustering routine, meaning two local clustering options with same performance measure. However, it is so possible that the clustering process is running toward a local optimum because lacking global information. In [6], a random walk through a graph is introduced to grab an idea on global structure to avoid bad local merging. Lacking the capability of exploring global structure of a graph, we incorporate in NNMCM algorithm a feature of allowing bad merging all the time to enable breakout from local optima. This idea is common in distributed coloring problem on conflict graphs. In the following, we will start by describing a simple distributed clustering algorithm without bad merging, a Greedy Distributed Minimum Cut (GDMC) algorithm. Then, the feature of allowing bad merging in NNMCM is elaborated.

In GDMC, each node executes periodically a routine. From the point of view of one node, we denote the time of executing the routine as active phase, and the rest of the period as passive phase. In active phase, a node can initiate changes to local clustering, and is denoted as an initiator. On the other hand, a node can only react to invitations for changes in passive phase. A node in active and passive phases of GDMC works as follows.

Active Phase

1. If my cluster is full, select one cluster member as kick-out candidate at random.
2. Invite one of neighboring nodes to join my cluster at random.
3. Calculate my new intra-cluster metric by summing up all intra-cluster weights.

4. Exchange new intra-cluster metric with the neighboring node.
5. Calculate sum of intra-cluster metric, q_{new}, of clusters related to this node and the invited node.
6. If $q_{\text{new}} > q_{\text{old}}$, add the neighboring node and remove the kick-out candidate (and the kick-out node forms a single-member cluster temporarily) from my cluster (if any) . Otherwise, do nothing.

Passive Phase

– Upon receipt of an invitation.

1. Calculate intra-cluster metric of my cluster without me by summing up all intra-cluster weights.
2. Exchange intra-cluster metric with the neighboring node.
3. Calculate sum of intra-cluster metric, q_{new}, clusters related to this node and the node that sends out the invitation.
4. If $q_{\text{new}} > q_{\text{old}}$, accept the invitation. Otherwise, do nothing.

– Upon receipt of kick-out request, leave original cluster and form a single-member cluster.

A new decision can be made after the initiator exchanges interference metric with the invited node. It is noted that as both the initiator and the invited node have the same information after exchanging interference metric, the decision made is also consistent. However, the kick-out candidate has to be informed if the decision is to remove it from its current cluster.

From the last rule of active phase, the clustering based on GDMC is improving all the time. The fact that GDMC gets stuck in the global optimum is a desirable feature; once a good clustering is found, discarding that clustering does not make sense. It is also favored to look for improvement in each iteration for quick convergence. On the other hand, GDMC does get stuck in local optima from which the progress to the global optimum is unlikely.

The clustering output of GDMC is evidently affected by which nodes get to run the routine earlier. In other words, the ordering of nodes impact on the clustering results much. By setting up an ordering, one almost at the same time decides which local optimum the graph clustering will go. To avoid this, one of course can run the routine with different node orderings for many times and pick the best. However, this kind of search for optimal node ordering requires global mechanism to some degree, which apparently conflicts with our definition of distributed clustering.

To remedy being stuck in local optima, one should add the possibility of making bad decisions. However, simply adding such decisions does not work satisfactorily since then the global minimum would no longer be an absorbing state. We address this issue in NNMCM algorithm. It is the same as GDMC except the last step in active phase and the last step of upon receipt of an invitation in passive phase. These steps are modified as follows.

Active Phase

6. If $q_{\text{new}} > q_{\text{old}}$, add the neighboring node and remove the kick-out candidate (and the kick-out node forms a single-member cluster temporarily) from my

cluster. Otherwise, with probability p add the neighboring node and remove the kick-out candidate (and the kick-out node forms a single-member cluster temporarily) from my cluster, and with probability $1-p$ do nothing.

Passive Phase

– Upon receipt of an invitation.
  4. If $q_{\text{new}} > q_{\text{old}}$, accept the invitation. Otherwise, with probability p accept the invitation, and with probability $1-p$ do nothing.

If $p = 0$, the method corresponds exactly to GDMC. When $0 < p < 1$ however, the method is able to do bad decisions locally and end up in better local and global state than when only minimizing local interference.

In this paper, we use the following definition for p

$$p = \exp\left(\beta(q_{\text{new}} - q_{\text{old}})\right). \tag{4}$$

It is noted that in (4), p is a variable depending on metric difference $(q_{\text{new}} - q_{\text{old}})$ rather than a fixed number. The bigger the metric difference, the lower p is. The rationale behind this is that the smaller the metric difference $(q_{\text{new}} - q_{\text{old}})$ is, the more likely it is to jump over the barrier to find another optima. This idea has been applied in e.g., simulated annealing [8], although in a slightly different manner. Note that we have GDMC scheme when $\beta = \infty$.

The performance of NNMCM is comparable to that of GDMC with the important difference that NNMCM converges to the global optimum even in difficult cases when GDMC does not. The convergent speed of both algorithms are similar. Both of them converges within few iterations (e.g., around $10-20$ in our simulation cases) where one iteration is defined as the time it takes for each node in network to run the corresponding routines once. This indicates a linear scalability of complexity and network size.

4 Simulation Results

Here, we evaluate the efficiency and performance of NNMS algorithm to solve distributed interference graph clustering problem. The figure of merit is the interference density measure defined in (1). With CoMP transmission, only interference generated by nodes residing in same cluster could be canceled. Therefore, the interference density measure reflects a percentage of total interference that could be potentially eliminated. This is the figure of merit most pertinent for wireless network operation. Complete interference graphs in systems with 400 nodes in a rectangular 20×20 triangular grid have been constructed according to the principles of Section 2.3. Edge weights are calculated using (3). We assume a fixed limit on cluster size of 3.

Fig. 1 illustrates the relationship between converged density measure Q and parameter β for one realization of interference graph. A higher β indicates a lower probability of making bad decisions. With very small β, bad decisions are taken all the time and thus, disappointing Q are observed. Density measure Q is improved as β increases, until it hits the maximum achieved by β value around 5. It is noted that Q converges to a constant for $\beta > 10$. When $\beta > 10$, the

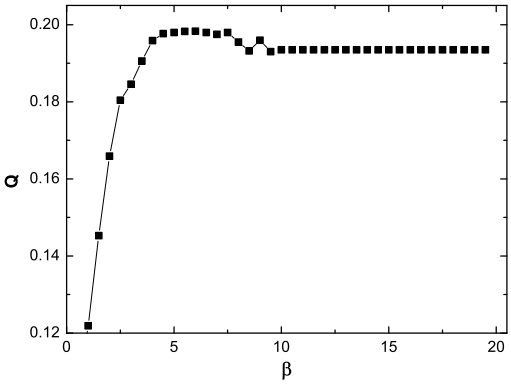


Fig. 1. Density measure Q with respect to β in NNMCM algorithm

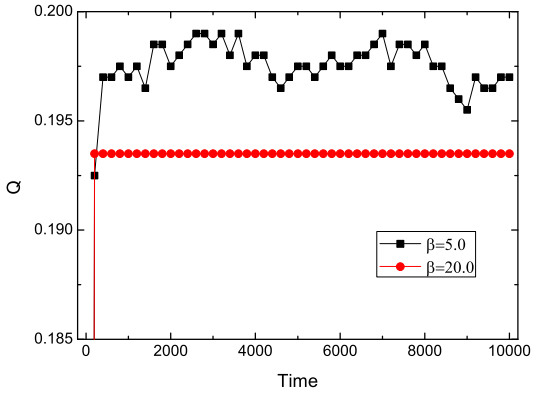


Fig. 2. Illustration of convergence time for NNMCM($\beta = 5.0$) and GDMC($\beta = 20.0$) algorithms

probability of taking bad decisions is low enough so that it does not affect the results in sensible way, and therefore can essentially be treated as the same as taking $\beta = \infty$.

Fig. 2 presents convergence behavior for NNMCM and GDMC algorithms. We use $\beta = 5.0$ and $\beta = 20.0$ for NNMCM and GDMC algorithms, respectively. The selected β values is justified by Fig. 1. The convergence time of both algorithms are comparable, with NNMCM showing better results due to its capability of escaping from local optima. Using GDMC, not any kind of random walk among local optima is possible, so that the whole network converges to a local optimum quickly and never escape from it. It is noted that the time unit here is the time period it takes for all nodes to run the corresponding routine one times for the whole network.

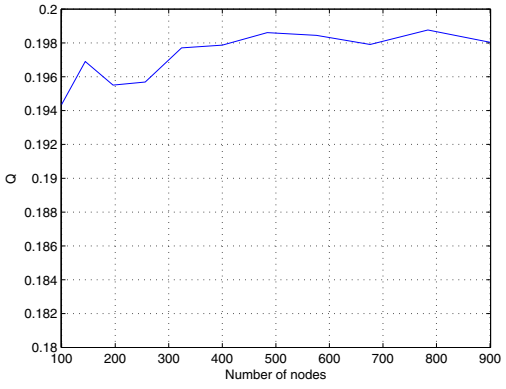


Fig. 3. Density measure Q with respect to network size in NNMCM algorithm. For each network size, optimized β value has been used.

Fig. 3 shows the relation between performance of NNMCM and network size. For each network size, the best numerically-searched β value is used. From Fig. 3, the converged density measure Q is stable with respect to network size, indicating NNMCM algorithm is robust to network scalability.

5 Conclusions

In this paper, we studied weighted graph clustering in a distributed manner. We have attempted to find the minimum cut clustering in a distributed manner, with restriction on cluster size. Due to a stated application scenario in CoMP transmission of wireless communication, the clustering algorithm is also required to provide a working clustering output in any moment.

The proposed Neighborhood Negotiation with Monte Carlo Method (NNMCM) clustering is capable of escaping from local optima by allowing a bad decision-making mechanism. The probability of making bad decision is adaptive to the depth of current local optimum. Our results show comparable efficiency of NNMCM to a Greedy Distributed Minimum Cut (GDMC) clustering where no local optima escaping mechanism is available. In addition, NNMCM algorithm outputs clustering results which show higher interference density measure than GDMC clustering. A linear scalability between complexity and network size is also featured in NNMCM.

For practical applications in CA formation problems, the proposed distributed clustering problem should be evaluated in a dynamic network where part of the entries of the interference matrix between nodes may change from time to time. In such a dynamic environment, another important issue would be how fast can the clustering method adapts to a good enough clusterization. In addition, it is likely that there are multiple radio resource shifts in the network, with each of the resource shift requiring an individual CA solution. Mobile stations in one cell need to be grouped to different resource shifts before the interference matrices

for different resource shifts can be constructed. This indicates the CA formation is coupled with the grouping of mobile stations, which is a challenging task. This is left for future work.

References

1. 3GPP: Feasibility study for further advancements of E-UTRA (LTE-Advanced). Tech. Rep. TR 36.912 (2009)
2. Clauset, A.: Finding local community structure in networks. Phsical Review E 72(2), 26132 (2005)
3. Deosarkar, B., Yadav, N., Yadav, R.: Clusterhead selection in clustering algorithms for wireless sensor networks: A survey, pp. 1–8 (December 2008)
4. Elias, P., Feinstein, A., Shannon, C.E.: Note on maximum flow through a network. IRE Transactions on Information Theory IT-2, 117–119 (1956)
5. 3rd Generation Partnership Project; TSG RAN: Physical layer aspects for evolved universal terrestrial radio access (UTRA). Tech. Rep. TR 25.814 (September 2006)
6. Harel, D., Koren, Y.: On clustering using random walks. In: Hariharan, R., Mukund, M., Vinay, V. (eds.) FSTTCS 2001. LNCS, vol. 2245, pp. 18–41. Springer, Heidelberg (2001)
7. Hopcroft, J., Khan, O., Kulis, B., Selman, B.: Natural communities in large linked networks. In: KDD 2003: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 541–546. ACM, New York (2003)
8. Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C.: Optimization by simulated annealing: an experimental evaluation. part i, graph partitioning. Oper. Res. 37, 865–892 (1989)
9. Ford Jr., L., Fulkerson, D.: Maximum flow through a network. Canadian Journal of Mathematics 8, 399–404 (1956)
10. Karakayali, M.K., Foschini, G.J., Valenzuela, R.A.: Network coordination for spectrally efficient communications in cellular systems. In: IEEE Wireless Commun., pp. 56–61 (August 2006)
11. Kernighan, B.W., Lin, S.: An Efficient Heuristic Procedure for Partitioning Graphs. The Bell System Technical Journal 49(1), 291–307 (1970)
12. Kulkarni, V., Forster, A., Venayagamoorthy, G.: Computational intelligence in wireless sensor networks: A survey. Communications Surveys Tutorials, IEEE PP(99), 1–29 (2010)
13. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Physical Review E 69, 066133 (2004)
14. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E 69(026113) (2004)
15. Schaeffer, S.E.: Stochastic local clustering for massive graphs. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 354–360. Springer, Heidelberg (2005)
16. Schaeffer, S.E.: Graph clustering. Computer Science Review 1, 27–64 (2007)
17. Stüber, G.L.: Principles of Mobile Communication (1996)
18. Yu, J., Chong, P.: A survey of clustering schemes for mobile ad hoc networks. IEEE Communications Surveys Tutorials 7(1), 32–48 (2005)

Considerations on Quality Metrics for Self-localization Algorithms

Juergen Eckert[1], Felix Villanueva[2], Reinhard German[1], and Falko Dressler[1]

[1] Computer Networks and Communication Systems, Dept. of Computer Science, University of Erlangen, Martensstrasse 3, 91058 Erlangen, Germany
{juergen.eckert,german,dressler}@cs.fau.de
[2] Computer Architecture and Networks, School of Computer Science, University of Castilla-La Mancha, Paseo de la Universidad 4, 13071 Ciudad Real, Spain
felix.villanueva@uclm.es

Abstract. The demand for location awareness and, therefore, the demand for self-localization techniques is continuously increasing. As a result, a good number of systems and methods for self-localization have been developed. Almost every system described in the literature exploits specific hardware or scenario features to solve the positioning issue, e.g. by using anchor nodes, relying on distances or angles, and even focusing on quite different distances ranging from centimeters to several kilometers. In many cases, the metrics used to evaluate the localization quality have been chosen according to the scenario. In this paper, we thoroughly discuss the most frequently used metrics for evaluating the quality of self-localization techniques. According to our findings, careful handling of some commonly used metrics is strongly required. We further propose an area-based solution that is especially helpful to measure and to compare different localization systems, which only need exact localization in a local context independently from specific scenario or hardware requirements. In this paper, we try to shed light on the question how to compare those very different techniques. In particular, we suggest the use one of two attribute-independent metrics. The first one is a generalization of an already quite popular metric, the Global Energy Ratio (GER), and the latter, the Area Ratio (AR), is a new approach based on the covered area.

Keywords: Localization, quality metrics, sensor networks, self-organization.

1 Introduction

Location awareness and the related localization techniques have become one of the most important issues in the field of wireless networking and mobile computing. With the emergence of a new class of location-aware distributed applications, and with new developments in the area of massively distributed systems, localization techniques are even becoming a key factor influencing the quality of those

C. Bettstetter and C. Gershenson (Eds.): IWSOS 2011, LNCS 6557, pp. 104–115, 2011.

applications. Among others, Wireless Sensor Networks (WSNs) and Sensor and Actor Networks (SANETs) are in the main focus of self-localization, i.e. the capability of autonomously generating (relative) location information [2,4].

One of the biggest problems of providing this information is to generate a reference grid or a map in advance. No system is working without this very costly or complex effort. Pure manual generation is very costly whereas a (partial or full) autonomous generation tends to be erroneous. The distinction between pure localization clients and producers is in most cases not obvious. As the system is growing the producers are expanding it in relation to the previously established knowledge. If this initial step or some measurements are wrong then this resulting placement error is mostly propagated through large parts of the set-up.

There is a wide variety of those localization techniques available, using lateration, angulation, and scene analysis techniques [3,6,16]. Many of the algorithms described in the literature have been designed for a specific sensing method but do not perform well in other environments. Some of these solutions have even been defined in the context of a complete self-localization system. Obviously, each of the proposed localization techniques has been thoroughly evaluated – in many cases not relying on a commonly accepted metric. This makes it impossible to compare or makes it very hard to choose the appropriate localization technique for an application. To simplify the evaluation of different approaches it would here be useful to have a limited amount of metrics.

Recently, a number of different metrics have been proposed. Most of them are taking special hardware attributes, measurement distances, or connectivity ranges into account; others are not independent of the number of nodes. A survey of general problems of localization including the used metrics can be found in the book by Mao and Fidan [14].

In this paper, we try to cast light on the quality of the localization results. We first briefly introduce the most common metrics and discuss their advantages and drawbacks. The main contribution is the finding that some metrics produce misleading results for certain classes of algorithms or scenarios. Furthermore, we present an area-based metric that is especially useful when studying algorithms that mainly relay on the localization accuracy in a local context. Based on the provided recommendations, it is possible to select an adequate metric for a particular class of algorithms and scenarios.

The remainder of the paper is organized as follows. Section 2 briefly describes our application scenario and motivates the paper. In Section 3, we introduce and discuss selected quality metrics including our proposed AR. Finally, Section 4 concludes the paper.

2 Evaluation Setup

Our scenario is a completely self-organizing approach to create a stable reference grid similar to the work by Liu *et al.* [13] using autonomous robot systems. Such systems can explore unknown environments and, at the same time, span a reference grid for highly accurate indoor local. Figure 1 depicts the basis scenario,

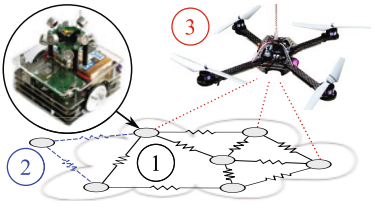


Fig. 1. Localization reference grid based on self-organizing robot systems

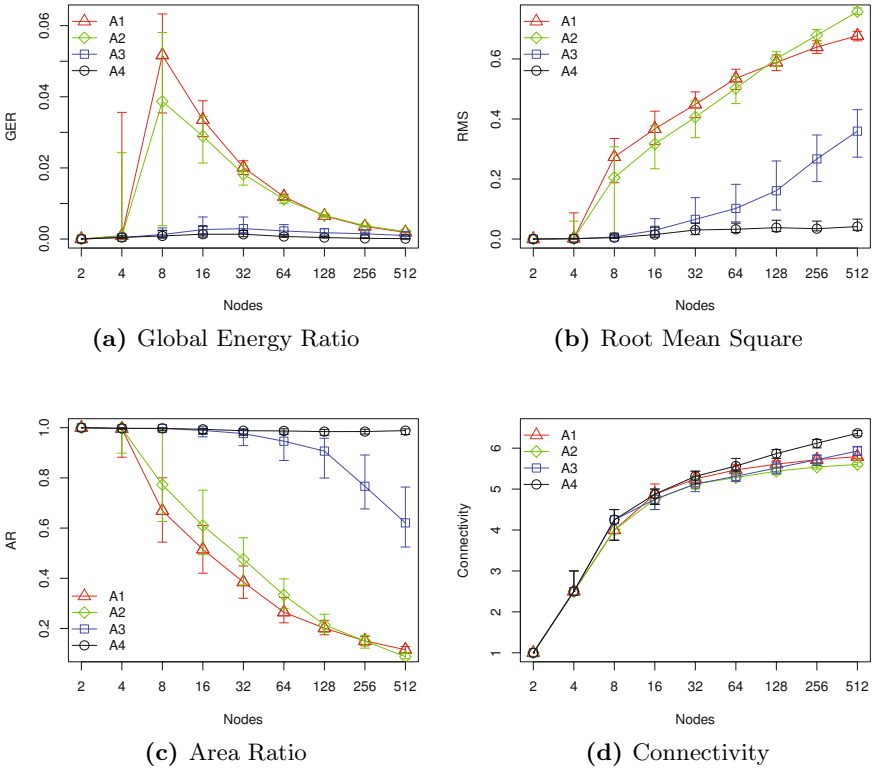
which we also used to describe our self-localization approach described in [5,6] Mobile nodes are autonomously deploying themselves on the ground forming a reference grid (step 1 in Figure 1). No global synchronization, knowledge, or state is available. Thus, nodes can arrive or depart from the system at any point in time (step 2 in Figure 1). A customer, e.g. a quadrocopter, can finally use the system to determine its current location (step 3 in Figure 1).

Our system relies on ultrasonic distance and (rough) angle measurements and supports accurate real-time localization of moving customers [7]. For more details on the scenario please refer to [8].

During the decision process of finding an appropriate method for creating and maintaining the reference system (step 1 in Figure 1), we observed that comparing different approaches is extremely challenging. Even the direct comparison of related algorithms is not possible in many cases: A lot of attributes have to be weighted against each other, accuracy and cost being the two primary ones.

Many researchers are creating their own evaluation scenario using hardware and scenario specific parameters, and are adapting some metric for the evaluation process. For comparison purposes, one would have to apply the very same parameters to a testbed, or would have to adjust the previous algorithms to work on the new hardware (or simulator). Both strategies may be very time consuming and even not be possible at all. Therefore, a common and widely accepted metric for comparison is needed, which is independent of any physical constrains such as the measurement technique, the network size, or the topology.

For the evaluation of different metrics, we implemented our fully distributed and self-organizing Mass-Spring-Relaxation (MSR) based technique [5,6] together with three different optimization steps of this algorithm in a simulation model. In the following, we use identifiers A1 to A4 to describe the different self-localization approaches: Algorithm A1 represents the classical MSR approach [11]. For Algorithm A2, variable node weights and an improved initial node positioning were introduced. Algorithm A3 features the capability of locally solving misplaced (flipped) node issues. Furthermore, the initial position is generated according to an existing network. Finally, Algorithm A4 introduces sector-of-arrival measurements in addition to the distance information. Details of these optimizations are out of the scope of this paper, please refer to [5,6]. As guideline, we can say that Algorithms A1 and A2 are performing rather poor, A3 performs quite good for small network sizes and, finally, Algorithm A4 performs nearly optimal for all tested network sizes.



(a) Global Energy Ratio **(b)** Root Mean Square

(c) Area Ratio **(d)** Connectivity

Fig. 2. Selected simulation results

As depicted in Figure 2, we simulated different network sizes of 2 to 512 nodes. For statistical evidence, we executed 200 runs for each experiment in randomly chosen topologies. The presented plots show the median (50 % quantile). The error bars depict the stable co-domain (25 % and 75 % quantiles). The details of Figures 2a, 2b and 2c will be explained later in detail.

Figures 2a, 2b, and 2c depict the variation in quality of the different refinements. The specific characteristics of the metrics will be discussed in Section 3. Figure 2d shows the average connectivity of a node for the different network sizes and optimization steps. It can be seen that all algorithms have the same connectivity. However, connectivity in general is an important attribute for most localization algorithms. After reaching convergence, this value becomes less important for evaluating the achieved position accuracy.

3 Towards a Common Quality Metric

We assume that the network the localization approaches are implemented and evaluated on is represented by an underlaying connected graph $G = (V, E)$ with

vertex set V and edge set E. Node i is associated with the vertex $N_i \in V$. Neighbors are defined as nodes that can directly interfere (sense and/or communicate) with each other. Such a node pair (i, j) is represented by an edge $(N_i, N_j) \in E$ in the graph. Two different mapping representations $p : V \rightarrow \Re^n$ for the underlying graph G exist. The first representation p_r is called *ground truth* and is defined as *a priori* knowledge of all physical node positions. Therefore it assigns a $\boldsymbol{x} \in \Re^n$ position tuple to a node ($\boldsymbol{x}_i = p_r(N_i)$). This knowledge is typically unknown to the algorithm but can easily be achieved within a simulator or within a controlled lab environment. The second representation p_v is called *virtual coordinate system*. Its position tuples $\widehat{\boldsymbol{x}} \in \Re^n$ represent the result of the algorithm ($\widehat{\boldsymbol{x}}_i = p_v(N_i)$).

Quality metrics in the field of localization techniques shall help to identify how accurate a certain algorithm is able to reconstruct the ground truth. With the exception of a few metrics (e.g., [9]), most of the existing metrics rely on this information. Those few metrics are, due to the lack of global knowledge, not so reliable, however, quite useful for a real application. In this paper, we assume that ground truth can be obtained.

Both *absolute* and *relative* metrics have to be considered. Absolute metrics compare the estimated with the physical positions:

$$p_r(N_i) = p_v(N_i) + \boldsymbol{\epsilon}_i;\ \forall N_i \in V \tag{1}$$

This is only applicable if anchors with known positions $p_v(\cdot)$ are present. These are acting as interfaces between the real and the information driven virtual world. Otherwise, the generated localization grid might be arbitrarily shifted, rotated, or mirrored. Absolute position metrics cannot be applied without those interconnecting elements. In contrast, relative metrics can always be applied to any system, because only relative relationships are necessary, e.g. ($\|\cdot\|_p$ is an arbitrary distance norm):

$$\|p_r(N_i) - p_r(N_j)\|_p = \|p_v(N_i) - p_v(N_j)\|_p + \epsilon_{i,j};\ \forall N_i, N_j \in V \tag{2}$$

A generalized quality metric should be able to rely on such relative positions.

3.1 Distance Based Metrics

In 2003, Priyantha et al. [15] introduced a metric called the Global Energy Ratio (GER). The authors removed all hardware-specific or technique dependent attributes by computing a normalized error $\widehat{e}_{ij}$ between the i-th and the j-th node according to Equation 3 ($(N_i, N_j) \in E_{all}; E \subseteq E_{all} := \{(N_a, N_b) | \forall N_a, \forall N_b \in V\}$; i.e. a fully connected graph), where the distances d_{ij} and $\widehat{d}_{ij}$ represent the true distance ($d_{ij} = \|p_r(N_i) - p_r(N_j)\|_2$) and the result of the algorithm ($\widehat{d}_{ij} = \|p_v(N_i) - p_v(N_j)\|_2$), respectively.

$$\widehat{e}_{ij} = \frac{\widehat{d}_{ij} - d_{ij}}{d_{ij}} \tag{3}$$

This error is then used for computing the GER as depicted in Equation 4, where N represents the network size.

$$\mathrm{GER} = \frac{\sqrt{\sum_{i,j:i<j} \widehat{e}_{ij}^2}}{N(N-1)\frac{1}{2}} \tag{4}$$

Figure 2a depicts the plot of GER in our simulation results. As the network size increases, the values are getting smaller. This is certainly not the expected behavior: The real errors get larger with the number of nodes. This indicates that the resulting error is not independent of the network size for GER.

In 2005, Ahmed et al. [1] published a new metric called the Global Distance Error (GDE) where they tried to compensate this defect. The same normalized error $\widehat{e}_{ij}$ (Equation 3) is used again. However, for computing the GDE according to Equation 5, a normalization is added according to the connectivity range R. Thereby, the authors took more than position into concern, which is again not applicable for arbitrary comparison, e.g. no distance ranges are available for angulation techniques. We omit the plots of our simulation results in this paper; the results are comparable to those shown in Figure 2b, scaled by $\frac{1}{R}$.

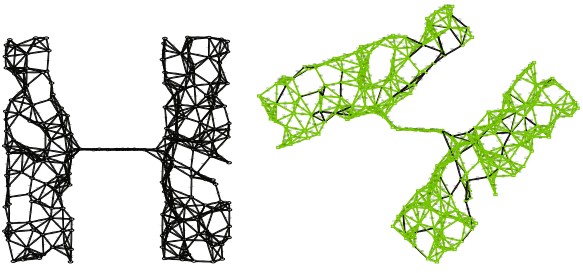
$$\mathrm{GDE} = \frac{1}{R}\sqrt{\frac{\sum_{i,j:i<j} \widehat{e}_{ij}^2}{N(N-1)\frac{1}{2}}} \tag{5}$$

In 2005, Gotsman and Koren [10] presented the Average Relative Deviation (ARD) metric. Here, each distance error is treated equally. The calculation of the ARD is depicted in Equation 6. In contrast, the GER explicitly aims to give more weight to outliers, so that errors become quickly obvious. Thus, this arithmetic mean is not very suitable for accurate error metrics – even though the resulting plots are looking pretty much the same as those of Figure 2b due to nearly uniform error distribution of our obstacle-less simulator.

$$\mathrm{ARD} = \frac{2}{N(N-1)} \sum_{i,j:i<j} \frac{\widehat{d}_{ij} - d_{ij}}{min(\widehat{d_{ij}}, d_{ij})} \tag{6}$$

Based on the described metrics and the observations made in our experiments, we suggest to utilize the Root Mean Square (RMS) of the normalized error, which is actually a minor variation of GER and GDE. The key issue was the sensitivity of the GER to the number of nodes, which is no longer present for RMS. In addition, the metric is more influenced by outliers (as the ARD) to make it more descriptive for worst cases. RMS also relies in the normalized errors $\widehat{e}_{ij}$ (Equation 3) and weights those compared to the number of nodes N in the network as shown in Equation 7. The plot in Figure 2b shows monotone increasing functions. This normalization regarding the network size allows to compare different approaches even if the network sizes are different. The RMS is meanwhile being frequently used for evaluating self-localization systems.

$$\mathrm{RMS} = \sqrt{\frac{\sum_{i,j:i<j} \widehat{e}_{ij}^2}{N(N-1)\frac{1}{2}}} \tag{7}$$



(a) **(b)**

Fig. 3. Simulation of an H-shaped network

3.2 Area Based Metrics

During our tests we made an interesting observation: Positioning errors are mainly located at the border of the network. Furthermore, they usually become detectable at the borders by deformations, e.g. parts of the network are folded in. This phenomenon is called *flip ambiguity* [12]. Nodes connect to previously flipped ones and propagate the error. The result are large virtually overlapping parts of the network which are causing large estimation errors. For a more detailed analysis of this major problem in localization please refer to the work of Kannan et al. [12]. Inside the network, the connectivity is higher and localization failures mainly accrue in areas with low connectivity. The probability for node misplacements inside of the system is low.

Analyzing localization systems described in the literature, we see that in most cases the deployment of sensor nodes is not relying on globally exact positioning of the individual nodes but on coverage. The exact distance estimation between two far away nodes (e.g., across the whole network) based on the virtual coordinate representation $p_v(\cdot)$ is mostly superfluous. Most location based services, as our client localization (step 3 in Figure 1), rely on accurate local position estimations and can absorb long range distance errors. The same holds for many other wireless applications. Thus, small local localization errors which may result in slight deformations of the virtual representation $p_v(\cdot)$ can be tolerated in most cases as long as the system makes sure that there is no overlapping of different network parts.

As an example, we investigated an H-shaped network as depicted in Figure 3 with a small bottleneck between two mass regions. Figure 3a shows the ground truth $p_r(\cdot)$. The bottleneck in this network is the very thin junction between the masses. Within this bottleneck, the probability for localization errors strongly increases. There is a high probability that this connection get bent in the virtual coordinate system $p_v(\cdot)$ as shown in Figure 3b. Table 1 shows the numerical results of this simulation.

Using the previously described distance metrics, the result shown in Figure 3b would be rated very poor. This happens, because these metrics synthesize a fully connected graph $G = (V, E_{all})$ and compare all the nodes with each other in the entire network regardless of their actual connectivity E. However, as can be seen,

Table 1. Metric results of Figure 3

Simulation of an H-shaped network	
Left side RMS:	3.659×10^{-4}
Right side RMS:	4.796×10^{-4}
Overall RMS:	1.204×10^{-1}
AR (convex hull):	96.38 %
AR (exact border):	98.73 %

the error is only high in a certain region. The overall RMS error for this example is 1.204×10^{-1} (in total, 220 nodes have been used in this example), which is for our algorithm and in comparison to Figure 2b at least one order of magnitude too high. Using this metric would thus result in discarding the result.

However, if both masses are evaluated individually, the resulting RMSs would be 3.659×10^{-4} and 4.796×10^{-4}, which are rather good results. The only problem is the bend in the middle. As stated before, in most cases, the following two conditions are relevant: First, the local accuracy for the localization service must be fulfilled (global accuracy, i.e. correct distance approximation of nodes far away from each other, is mostly negligible) and, secondly, parts of the network must not be overlapping to guarantee unique coordinates. The localization accuracy for the left as well as for the right masses are very good. Applications like location based routing or navigation in general work on the complete system as long as unique coordinates can be ensured. Based on the previous introduced metrics it is very had to get a clue whether parts of a network are overlapping or not.

In order to circumvent these problems, we introduce the Area Ratio (AR) as an area-based metric, which suits such non-uniform scenarios much better. AR brings both, the area of the ground truth A_r and the area of the generated grid A_v into relationship to each other as depicted in Equation 8.

$$\mathrm{AR} = \frac{A_v}{A_r} \tag{8}$$

The area of the covered real and the virtual ground can easily be computed by creating the convex hull over all nodes and, afterwards, computing the areas of the resulting polygons. We used the convex hull because our irregular simulation topologies were roughly formed like a round panel due to the absence of obstacles. Depending on the scenario, other techniques might be needed to (efficiently) find the borders of the network.

Figure 2c shows plots of our simulation results of the obstacle-less simulations. The values are more intuitive as from other metrics as they purely reflect the covered ground ratio. In this plot, differently to the previously plotted results for the distance based metrics, 1 (100 %) equals to perfect coverage.

Using the 220 node network shown in Figure 3b as an example, the AR using the convex hull for building the polygon gives an accuracy of 96.38 %. The convex

Table 2. Decision table for the appropriate metric approach

Network size	constant	variable		
Required distance accuracy	global	local (but unique)		global
Obstacles	any	w/o	w/	any
GER	X	-	-	-
RMS / GDE / ARD	X	O	-	X
AR (convex hull)	O	X	O	O
AR (exact border)	O	X	X	O

("X" → useable; "O" → use with care; "-" → do not use)

hull cannot fully cover the complexity of this specific shape, thus, the results might be too pessimistic. Using the exact borders of the shapes an accuracy of 98.73 % can be stated. Obviously, this value provides more realistic results compared to a distance based metrics such as the RMS metric.

The exact border of a representation p_x can for instance be found as follows. For each intersection of two edge $E_{i,j} := (N_i, N_j)$ and $E_{s,t}$ of a representation p_x a new vertex $N_{ij,st}$ is added to V. Its position $p_x(N_{ij,st})$ is located at the intersection of the edges. The original edges $E_{i,j}$ and $E_{s,t}$ get removed from E and new edges from N_i, N_j, N_s and N_t to $N_{ij,st}$ get inserted. Therefore in the representation p_x every intersection of edges gets removed. Subsequently an arbitrary vertex N_b located on the border of the graph representation p_x had to be found. The final polygon for the area computation can be found by following this border until vertex N_b is reached again.

4 Conclusions

In this paper, we discussed the need for comprehensive metrics for evaluating the quality of self-localization systems. We briefly reviewed the metrics available in the literature. First, we described a class of distance based metrics, the best known examples being GER and RMS. However, only the RMS provides the demanded capabilities for comparing the localization quality assuming different network sizes. We also discussed the quality and the problems of those metrics in general. In a second step, we presented an area-based metric that better fits the needs for estimating the localization quality in a local context.

Based on simulation results, we outlined the capabilities and limitations of these approaches. Depending on the application requirements, an appropriate evaluation metric has to be chosen. We conclude that the RMS metric should be used for evaluation if a global accuracy is required. If local accuracy is more important, e.g. for customer localization or information forwarding, results based on the RMS might be misleading. Instead, an area-based metric should be preferred. Table 2 summarizes our findings. It can be considered a conceptual matrix for selecting adequate metrics.

References

1. Ahmed, A.A., Shi, H., Shang, Y.: SHARP: A New Approach to Relative Localization in Wireless Sensor Networks. In: 25th IEEE International Conference on Distributed Computing Systems (ICDCS 2005), 2nd International Workshop on Wireless Ad Hoc Networking (WWAN 2005), pp. 892–898. IEEE, Columbus (2005)
2. Akyildiz, I.F., Kasimoglu, I.H.: Wireless Sensor and Actor Networks: Research Challenges. Elsevier Ad Hoc Networks 2, 351–367 (2004)
3. Bulusu, N., Estrin, D., Girod, L., Heidemann, J.: Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems. In: 6th International Symposium on Communication Theory and Applications (ISCTA 2001). Ambleside, Lake District (July 2001)
4. Dressler, F.: Self-Organization in Sensor and Actor Networks. John Wiley & Sons, Chichester (December 2007)
5. Eckert, J., Dressler, F., German, R.: Real-time Indoor Localization Support for Four-rotor Flying Robots using Sensor Nodes. In: IEEE International Workshop on Robotic and Sensors Environments (ROSE 2009), pp. 23–28. IEEE, Lecco (November 2009)
6. Eckert, J., German, R., Dressler, F.: An Indoor Localization Framework for Four-rotor Flying Robots Using Low-power Sensor Nodes. IEEE Transactions on Instrumentation and Measurement 60(2) (to appear, 2010)
7. Eckert, J., Koeker, K., Caliebe, P., Dressler, F., German, R.: Self-localization Capable Mobile Sensor Nodes. In: IEEE International Conference on Technologies for Practical Robot Applications (TePRA 2009), pp. 224–229. IEEE, Woburn (November 2009)
8. Eckert, J., Villanueva, F., German, R., Dressler, F.: A Self-Organizing Localization Reference Grid. In: 16th ACM International Conference on Mobile Computing and Networking (MobiCom 2010), Poster Session. ACM, Chicago (September 2010)
9. Girod, L.D.: A Self-Calibrating System of Distributed Acoustic Arrays. Phd thesis, University of California (December 2005)
10. Gotsman, C., Koren, Y.: Distributed Graph Layout for Sensor Networks. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 273–284. Springer, Heidelberg (2005)
11. Howard, A., Mataric, M.J., Sukhatme, G.: Relaxation on a Mesh: a Formalism for Generalized Localization. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001), Maui, HI, pp. 1055–1060 (October 2001)
12. Kannan, A.A., Fidan, B., Mao, G.: Analysis of Flip Ambiguities for Robust Sensor Network Localization. IEEE Transactions on Vehicular Technology 59(4), 2057–2070 (2010)
13. Liu, J., Zhang, Y., Zhao, F.: Robust Distributed Node Localization with Error Management. In: 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM Mobihoc 2006), Florence, Italy (May 2006)
14. Mao, G., Fidan, B. (eds.): Localization Algorithms and Strategies for Wireless Sensor Networks. Idea Group Inc. (IGI), USA (2009)
15. Priyantha, N.B., Balakrishnan, H., Demaine, E., Teller, S.: Anchor-Free Distributed Localization in Sensor Networks. Tech. Rep. TR-892, MIT Laboratory for Computer Science (April 2003)
16. Xiao, J., Ren, L., Tan, J.: Research of TDOA Based Self-localization Approach in Wireless Sensor Network. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006), pp. 2035–2040. IEEE, Beijing (October 2006)

A Plausibility of the AR Metric

It is clear that two different graph representations p_1, p_2 with the same vertex set V and edge set E that have equal areas are not necessarily identically looking. However, at least in the field of self-localization (where distances, positions, and/or angles are known), it holds that if two graph representations have similar areas, they are similarly shaped. Taking this into account, the AR (Equation 8) can only be meaningful if the following assumption holds for the entire connected graph as well as for the majority of its sub-graphs:

$$A_v \leq A_r \tag{9}$$

Under the following conditions we can show that Equation 9 is violated with a probability of $P_{violate} < 0.5$ for two neighboring triangles:

- If $\|p(N_a) - p(N_b)\|_2 < R$ then $(N_a, N_b) \in E$
- $\epsilon_{i,j}$ of Equation 2 is negligible small if $(N_i, N_j) \in E$
- No three connected nodes that are placed collinear

Exploding this entity and increasing the number of vertexes to infinity ($|V| \rightarrow \infty$ or a sufficiently large number), Equation 9 holds for the entire system.

Self-localization is usually a 2-dimensional problem. Thus, the area of a node N_a in which another node N_b can connect to is defined by:

$$A^A := \{\boldsymbol{x} \in \Re^2 \mid \|p(N_a) - \boldsymbol{x}\|_2 < R\} \tag{10}$$

Due to the described restriction, flipped areas can only occur if the connectivity of a newly joined node N_f is $d(N_f) = 2$. Let previously localized nodes be N_a and N_b. The possible placement $p(N_f)$ for the node N_f is within:

$$A^{AB} := \{\boldsymbol{x} \in \Re^2 \mid A^A \cap A^B\} \tag{11}$$

Now, let $N_c \in A^{AB}$ be a third node. As depicted in Figure 4, the nodes N_a, N_b, N_c are forming the triangle $\triangle ABC$. The virtual as well as truth graph representation ($p_v(\cdot)$ and $p_r(\cdot)$, respectively) must have the same areas. Without loss of generality we define the x-values of node N_a and node N_b to 0:

$$p(N_i) := (0, y_i)^T;\ i \in \{a, b\} \tag{12}$$

Thereby, two subsets A_+ and A_- representing the positive and the negative plane, respectively, are separated by the line $\overline{AB}$. A new node N_f can only be flipped if it is not connected to node N_c. This possible area is defined as:

$$A^{AB \backslash C} := \{\boldsymbol{x} \in \Re^2 \mid (A^A \cap A^B) \backslash A^C\} \tag{13}$$

A *bad* situation (the inverse of Equation 9) can only happen if for the ground truth representation $p_r(\cdot)$ the sign of the y-axis of the node N_c is the same as of node N_f ($sgn(y_f) = sgn(y_c)$). The resulting intersection of line $\overline{AC}$ and line $\overline{BF}$ is called D. If the virtual representation $p_v(N_f)$ for node N_f is flipped to $p_v(N_{f'})$

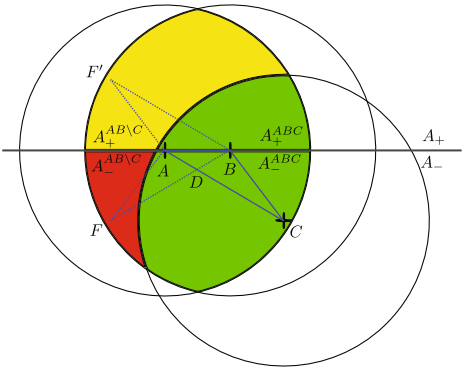


Fig. 4. Flip scenario

as depicted in Figure 4, Equation 9 does not hold. The area of the polygons can be calculated as follows:

$$A^{ABCDF} = A^{\triangle ABC} + A^{\triangle ABF} - A^{\triangle ABD} \tag{14}$$

$$A^{AF'BC} = A^{\triangle ABC} + A^{\triangle ABF'} \tag{15}$$

Note that $A^{\triangle ABF} = A^{\triangle ABF'}$. Thus, it can be easily seen that Equation 9 does not hold:

$$A_r = A^{ABCDF} < A^{AF'BC} = A_v \tag{16}$$

If we assume a worst case flip probability of $P_{flip} = 0.5$ (if no additional knowledge is present), then the probability of violating Equation 9 can be given as:

$$P_{violate} = \frac{A_-^{AB\backslash C}}{A^{AB\backslash C}} < 0.5 \tag{17}$$

Whereas $A_-^{AB\backslash C} = A^{AB\backslash C} \cap A_-$ and $A^{AB\backslash C} = A_+^{AB\backslash C} \cup A_-^{AB\backslash C}$ ($A_+^{AB\backslash C} \cap A_-^{AB\backslash C} = \emptyset$).

We now show that the following equation always holds:

$$A_-^{AB\backslash C} < A_+^{AB\backslash C} \tag{18}$$

Thus, the probability of placing a new node into $A_-^{AB\backslash C}$ (instead of $A_+^{AB\backslash C}$) is less then $P_{violate} < 0.5$. If this experiment is repeated for a sufficient number of nodes, the probability that Equation 9 hold is $\lim_{N\to\infty} P_{Eq.9} \to 1$.

Suppose $p_r(N_c) \in A_-$; $p_r(N_c) \in A_+$ can be shown analog. The area A^{AB} is symmetrical to the line $\overline{AB}$ which is equal to the x-axis. Thereby (due to the symmetry property):

$$A^C \cap A_- > A^C \cap A_+ \tag{19}$$

due to $p_r(N_c)_y < 0$. Thus:

$$A^{ABC} \cap A_- > A^{ABC} \cap A_+ \tag{20}$$

By migrating to the complimentary follows Equation 18 (*q.e.d.*).

Author Index