On the Computational Complexity of Gossip Protocols

Krzysztof R. Apt
CWI, The Netherlands
University of Warsaw, Poland
apt@cwi.nl

Eryk Kopczyński
University of Warsaw, Poland
erykk@mimuw.edu.pl

Dominik Wojtczak
University of Liverpool, UK
d.wojtczak@liv.ac.uk

Abstract

Gossip protocols deal with a group of communicating agents, each holding a private information, and aim at arriving at a situation in which all the agents know each other secrets. Distributed epistemic gossip protocols are particularly simple distributed programs that use formulas from an epistemic logic. Recently, the implementability of these distributed protocols was established (which means that the evaluation of these formulas is decidable), and the problems of their partial correctness and termination were shown to be decidable, but their exact computational complexity was left open. We show that for any monotonic type of calls the implementability of a distributed epistemic gossip protocol is a $\mathrm{P}_{||}^{\mathrm{NP}}$-complete problem, while the problems of its partial correctness and termination are in $\mathrm{coNP}^{\mathrm{NP}}$.

1 Introduction

The aim of this paper is to study natural complexity questions concerning gossip protocols. The set up of these protocols is the following. Each agent holds a secret initially known only to him. During the communications (for example phone calls) the participating agents share the secrets they know. The aim of the gossip protocols is to arrive at a situation in which all agents know all secrets. One of the early results established by a number of authors in the seventies, e.g., [Tijdeman, 1971], is that for $n \geq 4$ agents $2n-4$ phone calls is necessary and sufficient to reach such a final situation. However, the protocols solving the problem using $2n-4$ phone calls are centralized in the sense that they require a centralized scheduler. We are concerned here with the distributed gossip protocols that were introduced in [Attamah *et al.*, 2014b] and further studied with different type of calls in [Apt *et al.*, 2016]. These protocols use as guards epistemic formulas and thus are examples of ***knowledge based programs*** introduced in [Fagin *et al.*, 1997].

The formulation of distributed gossip protocols as knowledge-based programs considerably simplifies the task of their verification. The reason is that these protocols are defined simply as a parallel composition of simple loops in which the agents repeatedly evaluate a guard, which is an epistemic formula, and subsequently perform a corresponding call. As a result implementability of the protocol can be reduced to the problem of decidability of semantics of the underlying epistemic language and its partial correctness and termination to the problem of decidability of truth for this language.

In [Apt and Wojtczak, 2016] we established that such distributed epistemic gossip protocols (in short, protocols) are ***implementable***, i.e., the problem of evaluating a guard after a sequence of calls is decidable (implicitly shown there to be in EXPTIME), and that the problems of their partial correctness and termination are also decidable (implicitly shown there to be in 3-EXPTIME) in the setup when during the calls the agents exchange all their secrets (so-called push-pull type of calls) and the underlying topology of the network is a clique.

In this paper we sharpen this analysis and study the computational complexity of these three problems. We show that for any monotonic type of calls and network topology, the implementability of a protocol is an $\mathrm{P}_{||}^{\mathrm{NP}}$-complete problem and checking its partial correctness and termination is in $\mathrm{coNP}^{\mathrm{NP}}$.

Related work. Gossip protocols have been studied for more than forty years and have been successful in various domains, e.g., communication networks [Hedetniemi *et al.*, 1988], computation of aggregate information [Kempe *et al.*, 2003], and data replication [Ladin *et al.*, 1992]. A more recent account is given in the book [Hromkovic *et al.*, 2005] and in [Kermarrec and van Steen, 2007]. In these references gossip protocols are viewed as parallel, probabilistic and/or distributed programs.

Epistemic gossip protocols were studied in a number of recent publications. In [Attamah *et al.*, 2014a] a tool is discussed that given a high level description of an epistemic protocol in the setting of [Attamah *et al.*, 2014b] generates the characteristics of the protocol. The calls considered there differ from the ones considered here, so this approach is not applicable to our setting. In turn, in [van Ditmarsch *et al.*, 2017] dynamic distributed gossip protocols are studied in which the calls allow the agents not only to share the secrets but also to transmit the links. The purpose of the paper is to characterize such protocols in terms of the class of graphs for which they terminate. Such protocols then differ from the ones considered here, which are static. Next, [Herzig and Maffre, 2017] and [Cooper *et al.*, 2016b] consider gossip protocols that aim at achieving higher-order shared knowledge. Finally, in [Cooper *et al.*, 2016a] gossip protocols are expressed as an instance of multi-agent epistemic planning and subsequently translated into the classical planning language PDDL. In parallel with this work, we proved in [Apt and Wojtczak, 2017b] the decidability of fair termination for gossip protocols, while in [Apt and Wojtczak, 2017a] established first results concerning their extension with the common knowledge operator.

Plan. The paper is organized as follows. In the next two sections we recall the syntax and semantics of the gossip protocols considered in [Apt *et al.*, 2016]. The set up is more general since a broader definition of a call is adopted. To illustrate the considered protocols we discuss in Section 5 a protocol over an undirected graph, together with its partial correctness and termination proofs.

Next, in Section 6, we show that the following problem is NP-complete:

> Can a given distribution of sets of secrets among the agents be the outcome of a sequence of calls?

This problem turns out to be crucial for the computational complexity analysis of the partial correctness and of termination that is carried out in Section 7. Finally, in Section 8 we discuss some open problems.

2 Syntax

2.1 Calls and Call Types

Throughout the paper we assume a fixed finite set A of at least three ***agents*** that is an implicit parameter in all considerations. We assume that at the beginning each agent holds exactly one ***secret*** and that there exists a bijection between the set of agents and the set of secrets. We denote by S the set of all secrets. One could consider other initial secret distributions and the set of secrets to be larger than the set of agents. This would not alter our results, but would make the notation harder to read, so we opted for this simpler set up. Our aim is to analyse what the agents know after a sequence of calls took place. So first we introduce different type of calls and then consider an epistemic language allowing us to refer to agents' knowledge.

A ***call type*** is a pair $\bowtie = (E_{\bowtie}, f_{\bowtie})$, where $E_{\bowtie} \subseteq \mathsf{A} \times \mathsf{A}$ and $f_{\bowtie} : 2^{\mathsf{A}} \times 2^{\mathsf{A}} \to 2^{\mathsf{A}} \times 2^{\mathsf{A}}$. Intuitively, $f_{\bowtie}$ is the transformer of the sets of secrets that are familiar to the caller and callee. Each ***call*** is a triple $x \bowtie y$, where agent x is the caller, agent y is the callee y, and $\bowtie$ is the type of the call. A call $x \bowtie y$ is ***feasible*** if $(x, y) \in E_{\bowtie}$. Calls are denoted by c, d. Abusing notation we write $x \in \mathsf{c}$ to denote that agent x is one of the two agents involved in the call c.

In other words, $(\mathsf{A}, E_{\bowtie})$ is a directed graph and its edges specify which agent can $\bowtie$-call another. In turn, $f_{\bowtie}$ specifies the outcome of such a call given the sets of secrets the caller and callee are familiar with. In [Apt *et al.*, 2016] the following call types were studied.

- *Push-pull* calls, written as $x \circ y$ or simply xy, where agents x and y exchange all their secrets. In this case we define $f_{\circ}(X, Y) := (X \cup Y, X \cup Y)$, where X and Y are, respectively, the set of secrets the caller and callee are familiar with before this call takes place.
- *Push* calls, written as $x \rhd y$, where only the caller x passes his secrets to the callee y. In this case we define $f_{\rhd}(X, Y) := (X, X \cup Y)$.
- *Pull* calls, written as $x \lhd y$, where only the caller x learns the secrets of the callee y. In this case we define $f_{\lhd}(X, Y) := (X \cup Y, Y)$.

In this paper we generalize the setting and allow the outcome of a call to be any polynomially computable function, f, which is also *monotonic*, i.e., no agent ever forgets his secrets. Formally, if $f(X, Y) = (X', Y')$ then $X \subseteq X'$ and $Y \subseteq Y'$.

An example of a call type captured by this definition is the one in which during each call each agent reveals only one secret (e.g., the least one in some ordering on S). Our results also hold for call types with non-deterministic outcomes as long as there are polynomially many of them. We opted for deterministic call types only to keep the definitions simple.

2.2 Epistemic Logic

We consider formulas in a simple epistemic language $\mathcal{L}$ defined by the following grammar:

$$\phi ::= F_a p \mid \neg\phi \mid \phi \wedge \phi \mid K_a \phi,$$

where $p \in \mathsf{S}$ and $a \in \mathsf{A}$. Each secret is viewed as a distinct constant. We denote the secret of agent a by A, the secret of agent b by B and so on and sometimes implicitly switch between an agent and its secret.

We read $F_a p$ as 'agent a is familiar with the secret p' and $K_a \phi$ as 'agent a knows that formula ϕ is true'. So $F_a p$ is an atomic formula, while $K_a \phi$ is a compound formula. In fact, all atomic formulas of $\mathcal{L}$ have the form $F_a p$.

In what follows we shall distinguish the following sublanguages of $\mathcal{L}$:

- $\mathcal{L}_0$, its propositional part, which consists of the formulas that do not use the K_a modalities;
- $\mathcal{L}_1$, which consists of the formulas without the nested use of the K_a modalities;
- $\mathcal{L}_1^a$, where $a \in \mathsf{A}$ is a fixed agent, which consists of the formulas from $\mathcal{L}_1$ where the only modality is K_a.

3 Semantics

We now recall from [Apt *et al.*, 2016] semantics of the epistemic formulas. To this end we recall first the concept of a gossip situation.

3.1 Gossip Situations

A ***gossip situation*** (in short a ***situation***) is a sequence $\mathsf{s} = (\mathsf{Q}_a)_{a \in \mathsf{A}}$, where $\mathsf{Q}_a \subseteq \mathsf{S}$ for each agent a. Intuitively, Q_a is the set of secrets agent a is familiar with in situation s. The ***initial gossip situation*** is the one in which each Q_a equals $\{A\}$ (recall that A is the secret of agent a) and is denoted by root. This situation reflects the fact that initially each agent is familiar only with his own secret. We say that an agent a is an ***expert*** in a situation s if he is familiar in s with all the secrets, i.e., if $\mathsf{Q}_a = \mathsf{S}$.

We will use the following concise notation for gossip situations. Sets of secrets will be written down as lists. e.g., the set $\{A, B, C\}$ will be written as ABC. Gossip situations will be written down as lists of lists of secrets separated by dots. E.g., if there are three agents, then $\mathsf{root} = A.B.C$ and the gossip situation $(\{A, B\}, \{A, B\}, \{C\})$ will be written as $AB.AB.C$.

Each feasible call transforms the current gossip situation by modifying the set of secrets the agents involved in the call are familiar with. Consider a gossip situation $\mathsf{s} := (\mathsf{Q}_d)_{d \in \mathsf{A}}$. Then $(a \bowtie b)(\mathsf{s}) := (Q'_d)_{d \in \mathsf{A}}$, where $(\mathsf{Q}'_a, \mathsf{Q}'_b) = f_{\bowtie}(\mathsf{Q}_a, \mathsf{Q}_b)$, and $\mathsf{Q}'_c = \mathsf{Q}_c$, for $c \notin \{a, b\}$. This simply says that the only effect of a feasible call $a \bowtie b$ is that the secrets of the involved agents, a and b, are shared according to $f_{\bowtie}$.

3.2 Call Sequences

In [Apt *et al.*, 2016] computations of the gossip protocols were studied, so both finite and infinite call sequences were used. Here we focus on the finite call sequences as we are only interested in the semantics of epistemic formulas. So to be brief, unless explicitly stated, a ***call sequence*** is assumed to be finite. A call sequence is ***valid*** if each of its calls is feasible. Checking whether a call sequence is valid can easily be done in linear time, so from now on we assume that all considered call sequences are valid.

The empty call sequence is denoted by ϵ. We use $\mathbf{c}$ to denote a call sequence and $\mathbf{C}$ to denote the set of all finite (valid) call sequences. Given call sequences $\mathbf{c}$ and $\mathbf{d}$ and a call c we denote by $\mathbf{c}.\mathsf{c}$ the outcome of adding c at the end of the sequence $\mathbf{c}$ and by $\mathbf{c}.\mathbf{d}$ the outcome of appending the sequences $\mathbf{c}$ and $\mathbf{d}$.

The result of applying a call sequence to a situation s is defined inductively by putting $\epsilon(\mathsf{s}) := \mathsf{s}$ and $(\mathsf{c}.\mathbf{c})(\mathsf{s}) := \mathbf{c}(\mathsf{c}(\mathsf{s}))$.

Example 1 Let $\mathsf{A} = \{a, b, c\}$. Consider the call sequence $(ac, b \triangleright c, a \triangleleft c)$ involving three different call types. It generates the following successive gossip situations starting from root: $A.B.C \xrightarrow{ac} AC.B.AC \xrightarrow{b \triangleright c} AC.B.ABC \xrightarrow{a \triangleleft b} ABC.B.ABC$. Hence $(ac, b \triangleright c, a \triangleleft c)(\mathsf{root}) = (ABC.B.ABC)$. □

3.3 Gossip Models and Truth

A gossip situation is a set of possible combinations of secret distributions among the agents. As calls progress in sequence from the initial gossip situation, agents may be uncertain about which one of such secrets distributions is the actual one. This uncertainty is captured by appropriate equivalence relations on the call sequences.

Definition 1 *A* **gossip model** *is a tuple* $\mathcal{M} := (\mathbf{C}, \{\sim_a\}_{a \in \mathsf{A}})$, *where each* $\sim_a \subseteq \mathbf{C} \times \mathbf{C}$ *is the minimal relation satisfying the following conditions:*

- $\epsilon \sim_a \epsilon$,
- *Suppose* $\mathbf{c} \sim_a \mathbf{d}$.
  - *(i) If* $a \notin \mathsf{c}$, *then* $\mathbf{c}.\mathsf{c} \sim_a \mathbf{d}$ *and* $\mathbf{c} \sim_a \mathbf{d}.\mathsf{c}$.
  - *(ii) If* $a \in \mathsf{c}$ *and* $\mathbf{c}.c(\mathsf{root})_a = \mathbf{d}.c(\mathsf{root})_a$, *then* $\mathbf{c}.\mathsf{c} \sim_a \mathbf{d}.\mathsf{c}$.

A gossip model with a designated call sequence is called a **pointed gossip model**.

For instance, by *(i)* we have $ab, bc \sim_a ab, bd$. But we do not have $bc, ab \sim_a bd, ab$ since $(bc, ab)(\mathsf{root})_a = ABC \neq ABD = (bd, ab)(\mathsf{root})_a$.

The following two properties of $\sim_a$ from [Apt *et al.*, 2016] will be used in the sequel.

Fact 1 *(i) Each* $\sim_a$ *is an equivalence relation.*
(ii) For all $\mathbf{c}, \mathbf{d} \in \mathbf{C}$ *if* $\mathbf{c} \sim_a \mathbf{d}$, *then* $\mathbf{c}(\mathsf{root})_a = \mathbf{d}(\mathsf{root})_a$.

Finally, we recall the definition of truth.

Definition 2 *Let* $(\mathcal{M}, \mathbf{c})$ *be a pointed gossip model with* $\mathcal{M} := (\mathbf{C}, (\sim_a)_{a \in \mathsf{A}})$ *and* $\mathbf{c} \in \mathbf{C}$. *We define the satisfaction relation* $\models$ *inductively as follows (clauses for Boolean connectives are as usual and omitted):*

$$(\mathcal{M}, \mathbf{c}) \models F_a p \quad \textit{iff} \quad p \in \mathbf{c}(\mathsf{root})_a,$$
$$(\mathcal{M}, \mathbf{c}) \models K_a \phi \quad \textit{iff} \quad \forall \mathbf{d} \textit{ s.t. } \mathbf{c} \sim_a \mathbf{d},\ (\mathcal{M}, \mathbf{d}) \models \phi.$$

Further

$$\mathcal{M} \models \phi \quad \textit{iff} \quad \forall \mathbf{c}\, (\mathcal{M}, \mathbf{c}) \models \phi.$$

When $\mathcal{M} \models \phi$ *we say that* ϕ **is true**. □

So a formula $F_a p$ is true whenever secret p belongs to the set of secrets agent a is familiar with in the situation generated by the designated call sequence $\mathbf{c}$ applied to the initial situation root. In turn, the knowledge operator is interpreted as is customary in epistemic logic, using the equivalence relations $\sim_a$.

4 Gossip Protocols

In [Apt *et al.*, 2016], as a follow up on [Attamah *et al.*, 2014b], distributed epistemic gossip protocols were studied. Their goal is to reach a gossip situation in which each agent is an expert. In other words, their goal is to transform a gossip situation in which the formula $\bigwedge_{a \in \mathsf{A}} F_a A$ is true into one in which the formula $\bigwedge_{a, b \in \mathsf{A}} F_a B$ is true. Let us recall their definition.

By a ***component program***, in short a ***program***, for an agent a we mean a statement of the form $*[[]_{j=1}^{m} \psi_j \to \mathsf{c}_j]$, where $m > 0$ and each $\psi_j \to \mathsf{c}_j$ is such that $\psi_j \in \mathcal{L}_1^a$ and $a \in \mathsf{c}_j$.

Given a formula $\psi \in \mathcal{L}_1^a$ and a call c, we call the construct $\psi \to \mathsf{c}$ a ***rule*** and refer in this context to ψ as a ***guard***. The symbol [] denotes a nondeterministic choice among the rules of a given agent, while $*$ denotes a repeated execution of the rules, one at a time, where each time a rule is selected whose guard is true. Finally, by a ***distributed epistemic gossip protocol***, in short a ***gossip protocol***, we mean a parallel composition of component programs, one for each agent.

Assume a gossip protocol P that is a parallel composition of the component programs $*[[]_{j=1}^{m_a} \psi_j^a \to \mathsf{c}_j^a]$, one for each agent $a \in \mathsf{A}$.

The ***computation tree*** of P is defined as the (possibly infinite) set $\mathbf{C}^P$ of (possibly infinite) call sequences $\mathbf{c} = \mathsf{c}_0, \mathsf{c}_1, \ldots, \mathsf{c}_n, \ldots$ such that:

- $\mathbf{C}^P$ is closed under prefixes,
- for any call sequence $(\mathsf{c}_0, \mathsf{c}_1, \ldots, \mathsf{c}_i, \mathsf{c}_{i+1})$ in it: for some a and $j \in \{1, \ldots, m_a\}$ we have $(\mathcal{M}, (\mathsf{c}_0, \ldots, \mathsf{c}_i)) \models \psi_j^a$ and $\mathsf{c}_j^a = \mathsf{c}_{i+1}$.

  In this case we say that a transition between $(\mathsf{c}_0, \mathsf{c}_1, \ldots, \mathsf{c}_i)$ and $(\mathsf{c}_0, \mathsf{c}_1, \ldots, \mathsf{c}_i, \mathsf{c}_{i+1})$ took place due to the execution of the rule $\psi_j^a \to \mathsf{c}_j^a$.

By a ***computation*** of a gossip protocol we mean a maximal rooted path in its computation tree. We say that the gossip protocol P is ***partially correct*** if for all finite computations $\mathbf{c}$ of P: $(\mathcal{M}, \mathbf{c}) \models \bigwedge_{a, b \in \mathsf{A}} F_a B$, i.e., if each agent is an expert in the gossip situation $\mathbf{c}(\mathsf{root})$. Note that $\mathbf{c}$ is a finite computation iff $(\mathcal{M}, \mathbf{c}) \models \bigwedge_{a\in \mathsf{A}} \bigwedge_{j=1}^{m_a} \neg\psi_j^a$. We call the formula $\bigwedge_{a\in \mathsf{A}} \bigwedge_{j=1}^{m_a} \neg\psi_j^a$ the ***exit condition*** of the gossip protocol P. So P is partially correct iff the implication

$$\bigwedge_{a\in\mathsf{A}}\bigwedge_{j=1}^{m_a} \neg\psi_j^a \to \bigwedge_{a,b\in\mathsf{A}} F_a B \qquad (1)$$

is true. We say furthermore that P ***terminates*** if all its computations are finite.

All gossip protocols studied in [Apt *et al.*, 2016] use as guards only formulas from $\mathcal{L}_1$, that is in a program for agent a only guards from $\mathcal{L}_1^a$ are used.

5 Example Gossip Protocol

To illustrate the power of gossip protocols suppose that the agents are nodes of an undirected connected graph (V, E) and that the calls can take place only between agents connected by an edge. Let N_i denote the set of neighbours of node i.

Consider a gossip protocol with the following program for agent i:

$$*[[]_{j\in N_i, C\in \mathsf{S}} F_i C \wedge \neg K_i F_j C \to (i,j)].$$

Informally, agent i calls a neighbour j if i is familiar with some secret (here C) and he does not know whether j is familiar with it.

To prove its partial correctness consider the exit condition $\bigwedge_{(i,j)\in E} \bigwedge_{C\in\mathsf{S}} (F_i C \to K_i F_j C)$. For all agents i and j and secrets C, the formula $K_i F_j C \to F_j C$ is true, so the exit condition implies $\bigwedge_{(i,j)\in E} \bigwedge_{C\in\mathsf{S}} (F_i C \to F_j C)$.

Consider now an agent i and the secret J of agent j. Let $j = i_1, \ldots, i_h = i$ be a path that connects j with i. The above formula implies that for $g \in \{1, \ldots, h-1\}$ we have $\bigwedge_{C\in\mathsf{S}} (F_{i_g} C \to F_{i_{g+1}} C)$. By combining these $h-1$ formulas we get $\bigwedge_{C\in\mathsf{S}} (F_j C \to F_i C)$. But $F_j J$ is true, so we conclude $F_i J$. Consequently $\bigwedge_{i,j\in\mathsf{A}} F_i J$, as desired.

To prove termination it suffices to note that after each call ij the size of the set $\{(i,j,C) \mid \neg K_i F_j C\}$ decreases.

6 Incomplete Gossiping Problem

In [Apt and Wojtczak, 2016], the following problem was stated and was shown to be decidable.

Definition 3 (Incomplete Gossiping) *Can a given gossip situation* $\mathsf{s} = (Q_d)_{d\in\mathsf{A}}$ *be the outcome of a call sequence starting at* root*?*

Despite it apparent simplicity, we show that Incomplete Gossiping problem is NP-hard even if the only type of calls allowed is push-pull and everyone can call everybody else, i.e., the graph is a clique. This result is of independent interest as it connects with other computational problems. For instance, [Liben-Nowell, 2002] shows that computing the distance between two genomes can be reduced to the problem of computing the minimum number of calls necessary to reach a given gossip situation. In [Khuller *et al.*, 2003], the same was shown for the problem of data migration for storage devices in the setting where only one secret is exchanged during a call.

We show the stated result by a reduction from the following Triangle 3-Coloring problem.

Definition 4 (Triangle 3-Coloring) *The input is given as* (V, T) *where* $V = \{1, \ldots, n\}$ *be a set of vertices and* $T = \{(t_{1,0}, t_{1,1}, t_{1,2}), \ldots, (t_{m,0}, t_{m,1}, t_{m,2})\}$ *a set of* m *triangles, i.e., triplets of vertices. Let the set of colors be* $C = \{0, 1, 2\}$*.*

The problem is defined as follows. Is it possible to color each vertex in V *in such a way that each triangle is colored with three distinct colors, i.e., find a function* $c : V \to C$ *such that for each* $j \in T$, $k, l \in C$, $k \neq l$, *we have* $c(t_{j,k}) \neq c(t_{j,l})$*?*

Note that Triangle 3-Coloring problem is NP-complete – indeed, we can reduce from the standard NP-complete problem of Graph 3-coloring by adding a new fresh vertex to each edge, thus making it a triangle.

Theorem 1 *The* Incomplete Gossiping *problem is* NP*-hard.*

Proof. Let $\mathcal{I} = (V, T)$ be an instance of Triangle 3-Coloring. We will reduce it to an instance of Incomplete Gossiping $(\mathsf{A}_\mathcal{I}, \mathsf{s}_\mathcal{I})$, where $\mathsf{A}_\mathcal{I}$ is a set of agents and $\mathsf{s}_\mathcal{I}$ is a gossip situation. First, we will describe $(\mathsf{A}_\mathcal{I}, \mathsf{s}_\mathcal{I})$. Then we will show the intended call sequence $\mathbf{c}_\mathcal{I}$ and give some intuition about the gadgets that we use. We then show that if $\mathcal{I}$ is triangle 3-colorable, then $\mathbf{c}_\mathcal{I}(\mathsf{root}_{\mathsf{A}_\mathcal{I}}) = \mathsf{s}_\mathcal{I}$. In the last paragraph, we prove that our reduction is correct, i.e., if there exists $\mathbf{c}$ such that $\mathbf{c}(\mathsf{root}_{\mathsf{A}_\mathcal{I}}) = \mathsf{s}_\mathcal{I}$, then $\mathcal{I}$ is triangle-3-colorable.

Reduction. Let $\mathcal{I} = (V, T)$ be an instance of Triangle 3-Coloring. Let the set of agents be $\mathsf{A}_\mathcal{I} = \{C_{c,v} \mid c \in C, v \in V\} \cup \{A_{c,v} \mid c \in C, v \in V\} \cup \{F_c \mid c \in C\} \cup \{S_v \mid v \in V\} \cup \{K_{j,k} \mid j \in T, k \in C\} \cup \{L_{j,k} \mid j \in T, k \in C\} \cup \{G_v \mid v \in V\} \cup \{H_{j,k} \mid j \in T, k \in C\}$.

For every $c \in C, v \in V$ let us define $\text{color}(c) = \{C_{c,v} \mid v \in V\} \cup \{F_c\}$, $\text{Gadget}(v) = \{A_{c,v} \mid c \in C\} \cup \{S_v\}$, $\text{Fix} = \{F_c \mid c \in C\} \cup \{G_v \mid v \in V\} \cup \{H_{j,k} \mid j \in T, k \in C\}$, and $K(v) = \{K_{j,k} \mid t_{j,k} = v\}$. Also let $\text{AllColors} = \cup_{c\in C}\text{color}(c)$. The intuition behind the definition of these sets will become clearer once we define $\mathbf{c}_\mathcal{I}$ in the next subsection.

We now define the target gossip situation s as follows (where c iterates over C and v iterates V). Each agent $C_{c,v}$ is familiar with color(c) and Gadget(v). Each agent $A_{c,v}$ is familiar with color(c) and Gadget(v). Each agent F_c is familiar with AllColors and Fix. Each agent S_v is familiar with AllColors, Fix, Gadget(v), and $K(v)$. Each agent $K_{j,k}$ is familiar with AllColors, Fix, Gadget$(t_{j,k})$, $L_{j,k}$, and $K(t_{j,k})$. Each agent $L_{j,k}$ is familiar with AllColors, Gadget$(t_{j,l})$, $L_{j,l}$, and $K(t_{j,l})$ for all $l \in C$. Each agent G_v is familiar with AllColors, Fix, Gadget(v), and $K(v)$. Each agent $H_{j,k}$ is familiar with AllColors, Fix, Gadget$(t(j,k))$, $L_{j,k}$, and $K(t_{j,k})$.

Intended call sequence. We now define the call sequence $\mathbf{c}_\mathcal{I}$ such that $\mathbf{c}_\mathcal{I}(\mathsf{root}_{\mathsf{A}_\mathcal{I}}) = \mathsf{s}_\mathcal{I}$ if $\mathcal{I}$ is triangle-3-colorable.

1. At the beginning, for each $c \in C$, all agents in color(c) call each other. After this, for every agent $a \in \mathsf{A}_\mathcal{I}$, color $c \in C$, and at any point of $\mathbf{c}_\mathcal{I}$, agent a is either familiar with all the secrets from color(c), or none of them. We say that agent a ***has color*** $c \in C$ iff it is familiar with all the secrets from color(c).

2. In this step, we use a gadget which simulates coloring of the vertices with at most one color. For each $v \in V$, the gadget Gadget(v) consists of agents $A_{c,v}$ for each $c \in C$,

and the selection agent S_v. In step 2a, for each $v \in V$, we let S_v and all $A_{c,v}$ for $c \in C$ call each other. In step 2b, $A_{c,v}$ calls $C_{c,v}$. In step 2c, $A_{c,v}$ calls S_v.

Suppose that we request the agents $C_{c,v}$ and $A_{c,v}$ to be familiar with only secrets of Gadget(v) and color(c), and the agent S_v to be familiar with Gadget(v), but do not restrict its knowledge of secrets of color(c). It is, however, impossible for S_v to have more than one color at the end of this step, because otherwise one of the agents $C_{c,v}$ or $A_{c,v}$ would also have more than one color.

3. In this step, we make sure that each triangle is indeed colored with all three colors. In step 3a, we let agent S_v call all agents in $K(v)$, which is the set of all agents $K_{j,k}$ such that $t_{j,k} = v$. In step 3b, each agent $K_{j,k}$ calls $L_{j,k}$. In step 3c, for each $j \in T$, all agents $\{L_{j,k} \mid k \in C\}$ call each other, i.e., each agent $L_{j,k}$ will have all three colors.

4. The problem with the construction so far is that our gossiping situation at this point gives away the total knowledge of secrets of each agent. Currently, agents S_v and $K_{j,k}$ have just one color, which reveals the chosen coloring. To fix this, in step 4a, all agents in Fix call each other, thus all of them will have all colors. In step 4b, for each $v \in V$, agent G_v calls S_v, and, for each $j \in T$, $k \in C$, $H_{j,k}$ calls $K_{j,k}$.

Proof of correctness. Now, we will prove that $\mathsf{s}_\mathcal{I}$ can only be reached by a call sequence essentially the same as $\mathbf{c}_\mathcal{I}$.

W.l.o.g., we can assume that the call sequence has to start, for each color $c \in C$, with all agents in color(c) calling each other. If $\mathbf{c}$ is a call sequence such that $\mathbf{c}(\mathsf{root}_{\mathsf{A}_\mathcal{I}}) = \mathsf{s}_\mathcal{I}$ then adding these calls at the beginning of $\mathbf{c}$, would not affect its final gossip situation.

Note that a call between agents X and Y is possible only if X is familiar with the secret of Y, and Y is familiar with the secret of X. Furthermore, if agent X calls Y and Z, and Y is not familiar with secret Z, then we know that X cannot call Y after he has called Z. Based on this, we can deduce that only the following calls are possible in a call sequence $\mathbf{c}$ leading to $\mathsf{s}_\mathcal{I}$. Each agent $C_{c,v}$ can call color(c), then $A_{c,v}$. Each agent $A_{c,v}$ can call Gadget$(v) \setminus \{S_v\}$, then $C_{c,v}$, then S_v. Each agent F_c can call color(c), then Fix. Each agent S_v can call Gadget(v), then $K(v)$, then G_v. Each agent $K_{j,k}$ can call $S_{t_{j,k}}$ and $K_{t_{j,k}}$, then $L_{j,k}$, then $H_{j,k}$. Each agent $L_{j,k}$ can call $K_{j,k}$, then $L_{j,l}$. Each agent G_v can call Fix, then S_v. Each agent $H_{j,k}$ can call Fix, then $K_{j,k}$.

Note that all these calls actually take place in $\mathbf{c}_\mathcal{I}$.

Now, consider how $L_{j,k}$ could have received the colors. He could have received them from $L_{j,l}$ for $l \in C$. Now, $L_{j,l}$ could have received these colors from $K_{j,l}$, but before $K_{j,k}$ has called $L_{j,k}$ or $H_{j,k}$. Hence, $K_{j,l}$ could have received the colors only from S_v where $v = t_{j,l}$, but only before $S_{t_{j,l}}$ called $K(v)$ or G_v. Hence, S_v must have received the color from Gadget(v). Since no agent in Gadget(v) other than S_v has more than one color, S_v must have received at most one color at that time, c_v. Hence, $L_{j,k}$ could have learned only the colors $c_{t_{j,l}}$ for $l \in C$. Thus, if there exists call sequence $\mathbf{c}$ such that $\mathbf{c}(\mathsf{root}_{\mathsf{A}_\mathcal{I}}) = \mathsf{s}_\mathcal{I}$ then $\mathcal{I}$ has to be triangle-3-colorable. □

7 Computational Complexity

We now use the result of the previous section to determine the computational complexity of the implementability of the gossip protocols. We focus on a more general problem of determining the complexity of semantics for formulas with no nested modalities. We begin with the simplest case.

Lemma 1 *For any formula $\phi \in \mathcal{L}_0$ and a call sequence $\mathbf{c}$ checking whether $(\mathcal{M}, \mathbf{c}) \models \phi$ is in P.*

Proof.(sketch) We construct $\mathbf{c}(\mathsf{root})$ in polynomial time by a successive application of the calls in $\mathbf{c}$. We then simply check (in polynomial time) whether ϕ holds in $\mathbf{c}(\mathsf{root})$. □

Consider a call sequence $\mathbf{c}$. If for some prefix $\mathbf{c}_1.\mathsf{c}$ of $\mathbf{c}$, $\mathbf{c}_1(\mathsf{root}) = \mathbf{c}_1.\mathsf{c}(\mathsf{root})$, then we say that c is ***redundant*** in $\mathbf{c}$. We say that a call sequence $\mathbf{c}$ is ***redundant free*** if no call c from $\mathbf{c}$ is redundant in it. First, let us recall the following result that follows from Lemmas 1 and 2 in [Apt and Wojtczak, 2016].

Lemma 2 *Suppose that $\mathbf{c} := \mathbf{c}_1, \mathsf{c}, \mathbf{c}_2$ and $\mathbf{d} := \mathbf{c}_1, \mathbf{c}_2$, where c is redundant in $\mathbf{c}$. Then for all formulas $\phi \in \mathcal{L}_0$, $(\mathcal{M}, \mathbf{c}) \models \phi$ iff $(\mathcal{M}, \mathbf{d}) \models \phi$. Also, the maximum length of a redundant free call sequence is $|\mathsf{S}|^2$.*

We can now use the result from the previous section.

Lemma 3 *For any formula $\phi \in \mathcal{L}_0$, checking whether $\mathcal{M} \models \phi$ is* CO-NP*-complete.*

Proof. We consider the complement of this problem. By the definition of semantics and Lemma 2, $\mathcal{M} \not\models \phi$ holds iff for some redundant free call sequence $\mathbf{c}$, $(\mathcal{M}, \mathbf{c}) \models \neg\phi$. By Lemma 2 the length of such a call sequence is at most $|\mathsf{S}|^2$. This in conjunction with Lemma 1 implies that the complement problem is in NP.

To show NP-hardness note that each gossip situation $\mathsf{s} = (Q_d)_{d \in \mathsf{A}}$ can be encoded as the following conjunction of size at most $|\mathsf{S}|^2$: $\zeta(\mathsf{s}) = \bigwedge_{a \in \mathsf{A}} \left(\bigwedge_{B \in Q_a} F_a B \wedge \bigwedge_{B \notin Q_a} \neg F_a B \right)$. Now, a gossip situation s is a solution to the INCOMPLETE GOSSIPING problem iff $\exists \mathbf{c}\ (\mathcal{M}, \mathbf{c}) \models \zeta(\mathsf{s})$ iff $\mathcal{M} \not\models \neg\zeta(\mathsf{s})$. So the NP-hardness follows from Theorem 1. □

Lemma 4 *For any formula $\phi \in \mathcal{L}_0$ and a call sequence $\mathbf{c}$ checking whether $(\mathcal{M}, \mathbf{c}) \models K_a\phi$ is* CO-NP*-complete.*

Proof. By definition $(\mathcal{M}, \mathbf{c}) \models K_a\phi$ holds iff $\forall \mathbf{d}$ s.t. $\mathbf{c} \sim_a \mathbf{d}$, $(\mathcal{M}, \mathbf{d}) \models \phi$. Due to Lemma 2, it suffices to consider only call sequences $\mathbf{d}$ in which all calls except those involving agent a are redundant. The same argument as in the proof of Lemma 2 shows that the length of each such a call sequence $\mathbf{d}$ is at most $|\mathbf{c}| + |\mathsf{S}|^2$. This implies by Lemma 3 that checking $(\mathcal{M}, \mathbf{c}) \models K_a\phi$ can be done in CO-NP.

Now we show CO-NP-hardness already for the special case when $\mathbf{c} = \epsilon$ and ϕ does not refer to agent a. Note that $\mathbf{d} \sim_a \epsilon$ iff no call in $\mathbf{d}$ involves agent a. Consider the model $\mathcal{M}'$ with the set of agents $\mathsf{A}' = \mathsf{A} \setminus \{a\}$. Notice that $\forall \mathbf{d} \sim_a \epsilon\ (\mathcal{M}, \mathbf{d}) \models \phi$ iff $\forall \mathbf{d}\ (\mathcal{M}', \mathbf{d}) \models \phi$ iff $\mathcal{M}' \models \phi$. The conclusion now follows from Lemma 3. □

We can now establish an appropriate complexity result that refers to the already mentioned complexity class $\mathrm{P}^{\mathrm{NP}}_{\|}$.

Theorem 2 *For any formula $\psi \in \mathcal{L}_1$ and a call sequence $\mathbf{c}$ checking whether $(\mathcal{M}, \mathbf{c}) \models \psi$ is $P^{NP}_{\|}$-complete.*

Proof. We first show that the problem is in $\mathrm{P}^{\mathrm{NP}}_{\parallel}$, i.e., solvable by a deterministic polynomial-time Turing machine with parallel (i.e., non-adaptive) access to an NP oracle. Fix the appropriate formula ψ and a call sequence $\mathbf{c}$. By assumption ψ is a propositional formula over the set of basic formulas of the form $F_a B$ or $K_a\phi$, where ϕ is a propositional formula. We first check for each subformula $K_a\phi$ of ψ whether $(\mathcal{M}, \mathbf{c}) \models K_a\phi$. By Lemma 4 each such a check can be done by a single query to an NP oracle.

Replace now in ψ each basic subformula ϕ by **true** if $(\mathcal{M}, \mathbf{c}) \models \phi$ and by **false** otherwise. The resulting ground propositional formula is true iff $(\mathcal{M}, \mathbf{c}) \models \psi$. The former problem is in P, so the latter is in $\mathrm{P}^{\mathrm{NP}}_{\parallel}$.

To prove $\mathrm{P}^{\mathrm{NP}}_{\parallel}$-hardness we show a reduction from the following decision promise problem.

Definition 5 (Odd-Index) *Let $\bar{\mathsf{s}} = \mathsf{s}_1, \ldots, \mathsf{s}_k$ be a sequence of gossip situations such that for some $i \leq k$ all $\mathsf{s}_1, \ldots, \mathsf{s}_i$ can be the outcome of a call sequence (i.e., are positive instances of* Incomplete Gossiping*) and none of $\mathsf{s}_{i+1}, \ldots, \mathsf{s}_k$ can. Decide whether this promised index i is an odd number.*

Since the Incomplete Gossiping problem is NP-hard, the $\mathrm{P}^{\mathrm{NP}}_{\parallel}$-hardness of the Odd-Index problem follows from the sufficient conditions for $\mathrm{P}^{\mathrm{NP}}_{\parallel}$-hardness given in Theorem 5.2 in [Wagner, 1987] (see also Lemma 7 in [Spakowski and Vogel, 2000]). Now given an Odd-Index instance $\bar{\mathsf{s}} = \mathsf{s}_1, \ldots, \mathsf{s}_k$ we construct a formula $\psi \in \mathcal{L}_1$ such that $(\mathcal{M}, \epsilon) \models \psi$ iff the index i for $\bar{\mathsf{s}}$ is odd. First, let us add an additional agent a to A and set his secrets in $\bar{\mathsf{s}}$ to $\{A\}$. For a gossip situations s, let $\zeta(\mathsf{s})$ be defined as in the proof of Lemma 3. Let ψ_l be the formula $\bigwedge_{j=1}^{l} \neg K_a \neg\zeta(\mathsf{s}_j) \wedge \bigwedge_{j=l+1}^{k} K_a\neg\zeta(\mathsf{s}_j)$. Intuitively, ψ_l is true iff the index i is equal to l. This is the case because $(\mathcal{M}, \epsilon) \models \neg K_a\neg\zeta(\mathsf{s})$ iff s can be the outcome of a call sequence, and $(\mathcal{M}, \epsilon) \models K_a\neg\zeta(\mathsf{s})$ iff s cannot be the outcome of a call sequence. Let $\psi = \psi_1 \vee \psi_3 \vee \ldots \vee \psi_{2\lceil k/2\rceil-1}$. We claim that $(\mathcal{M}, \epsilon) \models \psi$ iff i is an odd number. This is because ψ tests whether i is equal to an odd number between 1 and k. □

This yields the desired conclusion about the implementability of the gossip protocols. To analyze their partial correctness and termination, we will need the following three lemmas.

Lemma 5 *For any call sequences $\mathbf{c} \sim_a \mathbf{d}$ and arbitrary formula ϕ: $(\mathcal{M}, \mathbf{c}) \models K_a\phi$ iff $(\mathcal{M}, \mathbf{d}) \models K_a\phi$.*

Proof. It follows directly from the definition of K_a. □

We call the second call c in a call sequence $\mathbf{c}_1.\mathsf{c}.\mathbf{c}_2.\mathsf{c}.\mathbf{c}_3$ ***epistemically redundant*** if $\mathbf{c}_1.\mathsf{c}(\mathsf{root}) = \mathbf{c}_1.\mathsf{c}.\mathbf{c}_2.\mathsf{c}(\mathsf{root})$. We now show that removing an epistemically redundant call does not affect the truth of any formula with no nested modalities.

Lemma 6 *If $\mathbf{c}_1.\mathsf{c}.\mathbf{c}_2.\mathsf{c}.\mathbf{c}_3$ is a call sequence where the 2nd call c is epistemically redundant, then for any formula $\psi \in \mathcal{L}_1$:*

$$(\mathcal{M}, \mathbf{c}_1.\mathsf{c}.\mathbf{c}_2.\mathsf{c}.\mathbf{c}_3) \models \psi \textit{ iff } (\mathcal{M}, \mathbf{c}_1.\mathsf{c}.\mathbf{c}_2.\mathbf{c}_3) \models \psi.$$

Proof. The proof proceeds by structural induction and the only non-trivial case is when $\psi = K_a\phi$. If $a \not\in \mathsf{c}$ then $\mathbf{c}_1.\mathsf{c}.\mathbf{c}_2.\mathsf{c}.\mathbf{c}_3 \sim_a \mathbf{c}_1.\mathsf{c}.\mathbf{c}_2.\mathbf{c}_3$, because $\mathbf{c}_1.\mathsf{c}.\mathbf{c}_2.\mathsf{c}(\mathsf{root}) = \mathbf{c}_1.\mathsf{c}.\mathbf{c}_2(\mathsf{root})$. The claim then follows from Lemma 5.

If $a \in \mathsf{c}$, it suffices to check that

$$\forall \mathbf{c}_1'.\mathsf{c}.\mathbf{c}_2'.\mathsf{c}.\mathbf{c}_3' \sim_a \mathbf{c}_1.\mathsf{c}.\mathbf{c}_2.\mathsf{c}.\mathbf{c}_3$$
$$(\mathcal{M}, \mathbf{c}_1'.\mathsf{c}.\mathbf{c}_2'.\mathsf{c}.\mathbf{c}_3') \models \phi \text{ iff } (\mathcal{M}, \mathbf{c}_1'.\mathsf{c}.\mathbf{c}_2'.\mathbf{c}_3') \models \phi.$$

First, due to Fact 1, $\mathbf{c}_1'.\mathsf{c}.\mathbf{c}_2'.\mathsf{c}.\mathbf{c}_3' \sim_a \mathbf{c}_1.\mathsf{c}.\mathbf{c}_2.\mathsf{c}.\mathbf{c}_3$ implies that $\mathbf{c}_1'.\mathsf{c}(\mathsf{root}) = \mathbf{c}_1'.\mathsf{c}.\mathbf{c}_2'.\mathsf{c}(\mathsf{root})$. Thus, the second c is redundant in $\mathbf{c}_1'.\mathsf{c}.\mathbf{c}_2'.\mathsf{c}.\mathbf{c}_3'$ and the claim follows from Lemma 2. □

Lemma 7 *For any formula $\psi \in \mathcal{L}_1$, checking $\mathcal{M} \models \psi$ is in coNP^{NP}.*

Proof. By definition $\mathcal{M} \models \psi$ holds iff $\forall \mathbf{c}\ (\mathcal{M}, \mathbf{c}) \models \psi$. Due to Lemma 6 it suffices to check this condition for call sequences $\mathbf{c}$ of polynomial length, because there are polynomially many different calls c and due to Lemma 2 in each $\mathbf{c}$ at most $|S|^2$ calls may be non-redundant. At the same time $(\mathcal{M}, \mathbf{c}) \models \psi$ is $\mathrm{P}^{\mathrm{NP}}_{\parallel}$-complete. This immediately gives a $\mathrm{coNP}^{\mathrm{P}^{\mathrm{NP}}_{\parallel}}$ algorithm for our problem, because it suffices to check for all polynomially-long call sequences $\mathbf{c}$ whether $(\mathcal{M}, \mathbf{c}) \models \psi$ holds. However, a $\mathrm{coNP}^{\mathrm{P}^{\mathrm{NP}}_{\parallel}}$ Turing machine can can be simulated by a $\mathrm{coNP}^{\mathrm{NP}}$ Turing machine, because polynomially many $\mathrm{P}^{\mathrm{NP}}_{\parallel}$ queries can be replaced by polynomially many queries to an NP oracle. This concludes the proof that the problem is in $\mathrm{coNP}^{\mathrm{NP}}$. □

Due to Lemma 7 and the characterization of partial correctness as the truth of formula (1) we get the following.

Theorem 3 *Checking partial correctness of a gossip protocol is in coNP^{NP}.*

We conclude by addressing the termination problem.

Theorem 4 *Checking whether a gossip protocol always terminates is in coNP^{NP}.*

Proof.(sketch) Non-termination of a gossip protocol is equivalent to checking whether for one of its guards, ψ, there exists a call sequence $\mathbf{c}$ and a call c such that both $(\mathcal{M}, \mathbf{c}) \models \psi$ and $(\mathcal{M}, \mathbf{c}.\mathsf{c}) \models \psi$ hold. Due to Theorem 2 checking either of them can be done in $\mathrm{P}^{\mathrm{NP}}_{\parallel}$. But in fact checking whether they both hold at the same time is also in $\mathrm{P}^{\mathrm{NP}}_{\parallel}$, because we can execute all their non-adaptive queries to an NP oracle simultaneously. This also shows that termination is in $\mathrm{P}^{\mathrm{NP}}_{\parallel}$, because one can simply negate the result of these checks. Now, the same argument as at the end of the proof of Lemma 7 shows that the termination problem is in $\mathrm{coNP}^{\mathrm{NP}}$. □

8 Future Work

In this paper, we established the computational complexity of implementability of a gossip protocol and an upper bound on checking its partial correctness and termination, which we conjecture to be $\mathrm{coNP}^{\mathrm{NP}}$-complete problems. An interesting future work would be to study the same problems for gossip protocols with nested modalities or with a common knowledge operator. Another interesting issue is to study the synthesis of a distributed gossip protocol from epistemic specifications (see, e.g., [van der Meyden and Wilke, 2005]).

Acknowledgements

We thank André Hernich for useful discussions on some aspects of this work. This work was initiated at Autobóz'16 and partly supported by NCN grant 2014/13/B/ST6/01807 and EPSRC grants EP/M027651/1 and EP/P020909/1.

References

[Apt and Wojtczak, 2016] Krzysztof R. Apt and Dominik Wojtczak. On decidability of a logic of gossips. In *Proc. of the 15th European Conference on Logics in Artificial Intelligence (JELIA 2016)*, volume 10021 of *Lecture Notes in Computer Science*, pages 18–33. Springer, 2016.

[Apt and Wojtczak, 2017a] Krzysztof R. Apt and Dominik Wojtczak. Common Knowledge in a Logic of Gossips. In *Proc. of TARK 2017 (to appear)*. EPTCS, 2017.

[Apt and Wojtczak, 2017b] Krzysztof R. Apt and Dominik Wojtczak. Decidability of Fair Termination of Gossip Protocols. In *Proc. of the IWIL Workshop and LPAR Short Presentations*, volume 1, pages 73–85. Kalpa Publications, 2017.

[Apt *et al.*, 2016] Krzysztof R. Apt, Davide Grossi, and Wiebe van der Hoek. Epistemic protocols for distributed gossiping. In *Proc. of the 15th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2015)*, volume 215 of *EPTCS*, pages 51–66, 2016.

[Attamah *et al.*, 2014a] Maduka Attamah, Hans van Ditmarsch, Davide Grossi, and Wiebe van der Hoek. A framework for epistemic gossip protocols. In *Proc of the 12th European Conference on Multi-Agent Systems (EUMAS 2014)*, pages 193–209, 2014.

[Attamah *et al.*, 2014b] Maduka Attamah, Hans van Ditmarsch, Davide Grossi, and Wiebe van der Hoek. Knowledge and gossip. In *Proceedings of ECAI'14*. IOS Press, 2014.

[Cooper *et al.*, 2016a] Martin C. Cooper, Andreas Herzig, Faustine Maffre, Frédéric Maris, and Pierre Régnier. A simple account of multiagent epistemic planning. In *Proc. of ECAI 2016*, pages 193–201. IOS Press, 2016.

[Cooper *et al.*, 2016b] Martin C. Cooper, Andreas Herzig, Faustine Maffre, Frédéric Maris, and Pierre Régnier. Simple Epistemic Planning: Generalised Gossiping. In *Proc. of ECAI 2016*, pages 1563–1564. IOS Press, 2016.

[Fagin *et al.*, 1997] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. Knowledge-based programs. *Distributed Computing*, 10(4):199–225, 1997.

[Hedetniemi *et al.*, 1988] Sandra Mitchell Hedetniemi, Stephen T. Hedetniemi, and Arthur L. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349, 1988.

[Herzig and Maffre, 2017] Andreas Herzig and Faustine Maffre. How to share knowledge by gossiping. *AI Communications*, 30(1):1–17, 2017.

[Hromkovic *et al.*, 2005] Juraj Hromkovic, Ralf Klasing, Andrzej Pelc, Peter Ruzicka, and Walter Unger. *Dissemination of Information in Communication Networks - Broadcasting, Gossiping, Leader Election, and Fault-Tolerance*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2005.

[Kempe *et al.*, 2003] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 482–491. IEEE, 2003.

[Kermarrec and van Steen, 2007] Anne-Marie Kermarrec and Maarten van Steen. Gossiping in distributed systems. *Operating Systems Review*, 41(5):2–7, 2007.

[Khuller *et al.*, 2003] Samir Khuller, Yoo-Ah Kim, and Yung-Chun Justin Wan. Algorithms for data migration with cloning. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 27–36. ACM, 2003.

[Ladin *et al.*, 1992] Rivka Ladin, Barbara Liskov, Liuba Shrira, and Sanjay Ghemawat. Providing high availability using lazy replication. *ACM Transactions on Computer Systems (TOCS)*, 10(4):360–391, 1992.

[Liben-Nowell, 2002] David Liben-Nowell. Gossip is synteny: Incomplete gossip and the syntenic distance between genomes. *Journal of Algorithms*, 43(2):264–283, 2002.

[Spakowski and Vogel, 2000] Holger Spakowski and Jörg Vogel. Θ_2^p-completeness: A classical approach for new results. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 348–360. Springer, 2000.

[Tijdeman, 1971] Robert Tijdeman. On a telephone problem. *Nieuw Archief voor Wiskunde*, 3(XIX):188–192, 1971.

[van der Meyden and Wilke, 2005] Ron van der Meyden and Thomas Wilke. Synthesis of distributed systems from knowledge-based specifications. In *Proceedings 16th International Conference CONCUR 2005*, volume 3653 of *Lecture Notes in Computer Science*, pages 562–576. Springer, 2005.

[van Ditmarsch *et al.*, 2017] Hans van Ditmarsch, Jan van Eijck, Pere Pardo, Rahim Ramezanian, and François Schwarzentruber. Epistemic protocols for dynamic gossip. *Journal of Applied Logic*, 20:1–31, 2017.

[Wagner, 1987] Klaus W. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51(1-2):53–80, 1987.