

Propositional Gossip Protocols under Fair Schedulers

Joseph Livesey and Dominik Wojtczak

University of Liverpool, UK

{joseph.livesey, d.wojtczak}@liverpool.ac.uk

Abstract

Gossip protocols are programs that can be used by a group of agents to synchronize what information they have. Namely, assuming each agent holds a secret, the goal of a protocol is to reach a situation in which all agents know all secrets. Distributed epistemic gossip protocols use epistemic formulas in the component programs for the agents. In this paper, we study the simplest classes of such gossip protocols: propositional gossip protocols, in which whether an agent wants to initiate a call depends only on the set of secrets that the agent currently knows. It was recently shown that such a protocol can be correct, i.e., always terminates in a state where all agents know all secrets, only when its communication graph is complete. We show here that this characterization dramatically changes when the usual fairness constraints are imposed on the call scheduler used. Finally, we establish that checking the correctness of a given propositional protocol under a fair scheduler is a coNP-complete problem.

1 Introduction

1.1 Background and Motivation

Gossip protocols have the goal of spreading information through a network via point-to-point communications (which we refer to as calls). Each agent holds initially a secret and the aim is to arrive at a situation in which all agents know each other's secret. During each call the caller and callee exchange all secrets that they know at that point. Such protocols were successfully used in a number of domains, for instance communication networks [Hedetniemi *et al.*, 1988], computation of aggregate information [Kempe *et al.*, 2003], and data replication [Ladin *et al.*, 1992]. For a more recent account see [Hromkovič *et al.*, 2005] and [Kermarrec and van Steen, 2007].

One of the early results established by a number of authors in the 1970s, e.g., [Tijdeman, 1971], is that for n agents $2n - 4$ calls are necessary and sufficient when every agent can communicate with any other agent. When such a communication graph is not complete, $2n - 3$ calls may be needed [Bumby, 1981] and are sufficient for any connected communication

graph [Harary and Schwenk, 1974]. However, the protocols considered in these papers were centralized, i.e., a central planner was telling each agent who and when to call.

In [Attamah *et al.*, 2014] a dynamic epistemic logic was introduced in which gossip protocols could be expressed as formulas. These protocols rely on agents' knowledge and are distributed, so they are *distributed epistemic gossip protocols*. This also means that they can be seen as special cases of knowledge-based programs introduced in [Fagin *et al.*, 1997].

In [Apt *et al.*, 2016] a simpler modal logic was introduced that is sufficient to define these protocols and to reason about their correctness. This logic is interesting in its own right and was subsequently studied in a number of papers. In this paper, we are going to focus on its simplest propositional fragment.

Propositional gossip protocols are a particular type of epistemic gossip protocols in which all guards are propositional. This means that calls being made by each agent are dependent only on the secrets that the agent has had access to. Such protocols have many potential applications. They are particularly simple and quick to execute (as evaluation of a guard can be done in linear time), so they are well-suited even for small devices with limited memory and computational capabilities. They can, e.g., be used for synchronization of information in sensor networks or swarms of robots. They can also be viewed as a special tractable case of multi-agent planning.

During the execution of a protocol, a state can be reached where multiple calls are possible at the same time. Then a call scheduler would decide which call takes priority. The scheduler can be assumed to be demonic, meaning it picks the order of calls in a way such that the protocol fails or to maximize the number of calls made before termination. In distributed and concurrent systems, it is common to impose fairness constraints on schedulers without which even the classic Peterson's algorithm for mutual exclusion would not work. In the context of gossip protocols this may require, e.g., that an agent makes a call eventually if he wants to call (which we will call *agent-fairness*) or that a call is eventually made if it can be (*rule-fairness*).

In [Apt and Wojtczak, 2019], many challenging open problems about general as well as propositional gossip protocols were proposed. The open problems listed there regarding propositional gossip protocols were addressed in [Livesey and Wojtczak, 2021b] and [Livesey and Wojtczak, 2021a]. In the former it was shown that the communication graph (i.e.,

who can call who) of a correct propositional protocol has to be complete (when ignoring the edges' directions), addressing Problem 5 of [Apt and Wojtczak, 2019]. We revisit here this answer by looking at the communication graph classes of correct propositional protocols under fair schedulers. We show that in this new setting this characterization completely changes: for any undirected connected graph, as well as directed strongly connected graph, there exists a correct propositional protocol under both agent-fair and rule-fair schedulers. Finally, we establish that checking the correctness of a given protocol under such fair schedulers is a coNP-complete problem.

1.2 Related Work

Much work has been done on general epistemic gossip protocols. The various types of calls used in [Attamah *et al.*, 2014] and [Apt *et al.*, 2016] were presented in a uniform framework in [Apt *et al.*, 2018], where in total 18 types of communication were considered and compared w.r.t. their epistemic strength. In this paper we study the asynchronous communication setting where agents are not aware of the calls they do not participate in nor can they synchronize their actions precisely as there is no global clock. This is the most appropriate model to study in the context of distributed systems. In [Apt and Wojtczak, 2018], the decidability of the semantics of the gossiping logic and truth in this model was established for the limited fragment of the logic (namely, without nesting of modalities). The precise computational complexity of this fragment was further studied in [Apt *et al.*, 2017] and checking termination, partial correctness and correctness in this case was shown to be in coNP^{NP}. In [Apt and Wojtczak, 2017] it was shown that checking agent-fair and rule-fair termination and correctness for such protocols are decidable. Building on these results, [van Ditmarsch *et al.*, 2020] showed decidability of the full logic for various variants of the gossiping model.

Centralized gossip protocols were studied in [Herzig and Maffre, 2017; Cooper *et al.*, 2016]. These had the goal to achieve higher-order shared knowledge. Such problems were further studied in [Cooper *et al.*, 2018] with temporal constraints where each call is assigned a time interval when this call can only be made.

In [van Ditmarsch *et al.*, 2017a; van Ditmarsch and Kokkinis, 2017] the expected time of termination of several gossip protocols for complete graphs was studied.

Dynamic distributed gossip protocols were studied in [van Ditmarsch *et al.*, 2017b], in which the calls allow the agents to transmit the links as well as share secrets. Similar to this paper, it characterized different distributed epistemic protocols in terms of the (largest) class of graphs where each protocol is correct, i.e., where the protocol necessarily ends up with all agents knowing all secrets.

1.3 Plan of the Paper

We first go through the logic, originally defined in [Apt *et al.*, 2016], in Section 2, where we also define the notions of agent-fair and rule-fair schedulers. We then investigate the structure of feasible communication graphs for agent-fair and rule-fair correct propositional gossip protocols in Section 3. Finally, we study the computational complexity of agent-fair and rule-fair termination and correctness in Section 4.

2 Gossiping Logic

We recall here the framework of [Apt *et al.*, 2016], which we restrict to the propositional setting. We assume a fixed set G of $n \geq 3$ **agents** and stipulate that each agent holds exactly one **secret**, and that there exists a bijection between the set of agents and the set of secrets. We use it implicitly by denoting the secret of agent a by A , of agent b by B , etc. We denote by Sec the set of all secrets.

The propositional language \mathcal{L}_p is defined by the following grammar:

$$\phi ::= F_a S \mid \neg\phi \mid \phi \wedge \phi,$$

where $S \in \text{Sec}$ and $a \in G$. We will distinguish the following sublanguage \mathcal{L}_p^a , where $a \in G$ is a fixed agent, which disallows all F_b operators for $b \neq a$.

So $F_a S$ is an atomic formula, which we read as ‘agent a is familiar with the secret S ’. Note that in [Apt *et al.*, 2016], a compound formula $K_a \phi$, i.e., ‘agent a knows the formula ϕ is true’, was used. Dropping $K_a \phi$ from the logic simplifies greatly its semantics and the execution of a gossip protocol, while still being capable of defining a rich class of protocols. Below we shall freely use other Boolean connectives that can be defined using \neg and \wedge in a standard way. We shall use the following formula

$$\text{Exp}_i \equiv \bigwedge_{S \in \text{Sec}} F_i S,$$

that denotes the fact that agent i is an **expert**, i.e., he is familiar with all the secrets.

Each **call**, written as ab or sometimes $a \circ b$, concerns two different agents, the **caller**, a , and the **callee**, b . After a call, the caller and the callee learn each other’s secrets. Calls are denoted by c , d , etc.

In what follows we focus on call sequences. Unless explicitly stated each call sequence is assumed to be finite. The empty sequence is denoted by ϵ . We use \mathbf{c} to denote a call sequence and \mathbf{C} to denote the set of all finite call sequences. Given call sequences \mathbf{c} and \mathbf{d} and a call c we denote by $\mathbf{c}.c$ 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} . We say that \mathbf{c}' is an **extension** of a call sequence \mathbf{c} if for some call sequence \mathbf{d} we have $\mathbf{c}' = \mathbf{c}.\mathbf{d}$.

To describe what secrets the agents are familiar with, we use the concept of a **gossip situation**. It is a sequence $\mathbf{s} = (Q_a)_{a \in G}$, where $\{A\} \subseteq Q_a \subseteq \text{Sec}$ for each agent a . Intuitively, Q_a is the set of secrets a is familiar with in the gossip situation \mathbf{s} . The **initial gossip situation** is the one in which each Q_a equals $\{A\}$ and is denoted by root . It reflects the fact that initially each agent is familiar only with his own secret. Note that an agent a is an expert in a gossip situation \mathbf{s} iff $Q_a = \text{Sec}$.

Each call transforms the current gossip situation by modifying the sets of secrets the agents involved in the call are familiar with as follows. Consider a gossip situation $\mathbf{s} := (Q_d)_{d \in G}$ and a call ab . Then

$$ab(\mathbf{s}) := (Q'_d)_{d \in G},$$

where $Q'_a = Q'_b = Q_a \cup Q_b$, and for $c \notin \{a, b\}$, $Q'_c = Q_c$.

So the effect of a call is that the caller and the callee share the secrets they are familiar with.

The result of applying a call sequence to a gossip situation s is defined inductively as follows:

$$\epsilon(s) := s, (\mathbf{c}.\mathbf{c})(s) := \mathbf{c}(\mathbf{c}(s)).$$

Example 1. 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 a comma. e.g., if there are three agents, a, b and c , then $\text{root} = A, B, C$ and the gossip situation $(\{A, B\}, \{A, B\}, \{C\})$ will be written as AB, AB, C .

Let $G = \{a, b, c\}$. Consider the call sequence $ac.bc.ac$. It generates the following successive gossip situations starting from root :

$$\begin{aligned} A, B, C &\xrightarrow{ac} AC, B, AC \xrightarrow{bc} \\ &AC, ABC, ABC \xrightarrow{ac} ABC, ABC, ABC. \quad \square \end{aligned}$$

Definition 1. Consider a call sequence $\mathbf{c} \in \mathbf{C}$. We define the satisfaction relation \models inductively as follows:

$$\begin{aligned} \mathbf{c} \models F_a S &\text{ iff } S \in \mathbf{c}(\text{root})_a, \\ \mathbf{c} \models \neg\phi &\text{ iff } \mathbf{c} \not\models \phi, \\ \mathbf{c} \models \phi_1 \wedge \phi_2 &\text{ iff } \mathbf{c} \models \phi_1 \text{ and } \mathbf{c} \models \phi_2. \end{aligned}$$

So a formula $F_a S$ is true after the call sequence \mathbf{c} whenever secret S belongs to the set of secrets agent a is familiar with in the situation generated by the call sequence \mathbf{c} applied to the initial situation root . Hence $\mathbf{c} \models \text{Exp}_a$ iff agent a is an expert in $\mathbf{c}(\text{root})$.

By a **propositional component program**, in short a **program**, for an agent $a \in G$ we mean a statement of the form

$$*\left[\bigwedge_{j=1}^{m_a} \psi_j \rightarrow c_j\right],$$

where each $\psi_j \rightarrow c_j$ (which we will call a **rule**), is such that

- agent a is the caller in the call c_j ,
- $\psi_j \in \mathcal{L}_p^a$ (which we will call a **guard** of this rule),

and $m_a \geq 0$ is the number of rules agent a has. If the guard of a rule is true then the corresponding agent, call and the rule is said to be **active**.

Intuitively, $*$ denotes a repeated execution of the rules, one at a time, where each time non-deterministically an active rule is selected.

Consider a **propositional gossip protocol**, P , that is a parallel composition of the propositional component programs $*\left[\bigwedge_{j=1}^{m_a} \psi_j^a \rightarrow c_j^a\right]$, one for each agent $a \in G$.

The **computation tree** of P is a directed tree defined inductively as follows. Its nodes are call sequences and its root is the empty call sequence ϵ . Further, if \mathbf{c} is a node and for some rule $\psi_j^a \rightarrow c_j^a$ we have $\mathbf{c} \models \psi_j^a$, then $\mathbf{c}.c_j^a$ is a node that is a direct descendant of \mathbf{c} . Intuitively, the arc from \mathbf{c} to $\mathbf{c}.c_j^a$ records the effect of the execution of the rule $\psi_j^a \rightarrow c_j^a$ performed after the call sequence \mathbf{c} took place.

By a **computation** of a gossip protocol P we mean a maximal rooted path in its computation tree. In what follows we identify each computation with the unique call sequence it generates. We say that the gossip protocol P is **partially correct** if for all leaves \mathbf{c} of the computation tree of P , and all

agents a , we have $\mathbf{c} \models \text{Exp}_a$, i.e., if each agent is an expert in the gossip situation $\mathbf{c}(\text{root})$. We say furthermore that P **terminates** if all its computations are finite and say that P is **correct** if it is partially correct and terminates.

In [Attamah et al., 2014] the following correct propositional gossip protocol, called *Learn New Secrets* (LNS in short), for complete digraphs was proposed.

Example 2 (LNS protocol). *The following program is used by agent i :*

$$*\left[\bigwedge_{j \in G} \neg F_i J \rightarrow ij\right].$$

Informally, agent i calls agent j if agent i is not familiar with j 's secret.

We also consider two variants of fair termination, but we first have to recall the notion of fairness. An infinite computation is **rule-fair** (resp. **agent-fair**) if all rules (resp. agents) that are active after infinitely many prefixes of this computation are selected infinitely often. By default a finite computation is rule-fair and agent-fair. We say that a gossip protocol P **rule-fairly terminates** (resp. **agent-fairly terminates**) if all its rule-fair (resp. agent-fair) computations are finite. We do not define agent-fair nor rule-fair partial correctness, because they are equivalent to partial correctness. This is simply because partial correctness is a condition on finite computations only and every finite computation is agent-fair and rule-fair. A protocol which rule-fairly (agent-fairly) terminates and is partially correct is said to be **rule-fair (agent-fair) correct**.

The agents and possible calls of a given protocol can be thought of as nodes (agents) and edges (calls) of its (**directed communication graph**). In the next section, we are going to study for which classes of communication graphs an agent-fair or rule-fair correct propositional gossip protocol exists.

3 Fair and Correct Communication Graphs

We begin by showing that the scheduler's type has a huge impact on whether a gossip protocol works correctly. In particular we show that rule-fairness and agent-fairness are not equivalent.

Proposition 1. *There exists an agent-fair correct propositional gossip protocol which is not a correct propositional gossip protocol.*

Proof. Let us have three agents, a, b and c . The protocol only consists of a rule $\neg F_a C \rightarrow ab$ for agent a , and $F_b A \wedge \neg F_b C \rightarrow bc$ for agent b . Clearly this is not a correct propositional gossip protocol as it may never terminate, e.g., ab call can be repeated indefinitely. However, this is an agent-fair correct propositional gossip protocol. Note that the first call has to be ab . After that call, both ab and bc calls are active. Due to the agent-fairness of the scheduler used, eventually the bc call must take place, ensuring that both b and c become experts. After that, the only active call is ab , and once this occurs no calls are active and so the protocol terminates. Moreover, all agents are experts at that point. \square

Note that, straight from the definition, any agent-fair correct protocol is also rule-fair correct, because rule-fairness can only narrow down further the set of permitted computations. We now show that the opposite does not hold.

Proposition 2. *There exists a rule-fair correct propositional gossip protocol which is not an agent-fair correct propositional gossip protocol.*

Proof. Let us have three agents, a , b and c . The protocol only consists of a rule $\neg F_a C \rightarrow ab$ for agent a , and rules $F_b A \wedge \neg F_b C \rightarrow bc$, $F_b A \wedge \neg F_b C \rightarrow ba$ for agent b . This is not an agent-fair correct propositional gossip protocol. Its computation has to start with ab , at which point calls ab , ba and bc are active. Note that a computation repeatedly alternating between making calls ab and ba after that is agent-fair, because a call of each active agent (a and b only) is picked infinitely many times. The protocol will not terminate in this case, so it is not agent-fair correct. However, this is a rule-fair correct propositional gossip protocol, because under a rule-fair scheduler the active call bc has to be picked eventually. Once bc is made both b and c will become experts. This leaves ab as the only active call, and once this occurs, the protocol terminates in a gossip situation where all agents are experts. \square

With this knowledge that the type of the scheduler used can change if a given protocol is correct, we revisit the Problem 5 of [Apt and Wojtczak, 2019] to determine what the underlying communication graph should be for agent-fair or rule-fair correct propositional gossip protocols. We start with the simplest case of a path (linear) graph where the n agents are labeled x_1, \dots, x_n and, for any $i < n$, agent x_i can only call x_{i+1} . We then proceed with increasingly broader graph classes, which on the downside require increasingly more complex protocols to be agent-fair or rule-fair correct.

Theorem 1. *For any path graph, there exists an agent-fair correct propositional gossip protocol.*

Proof. The protocol will only consist of rules $\neg Exp_{x_1} \rightarrow x_1 x_2$ for agent x_1 , $F_{x_2} X_1 \wedge \neg Exp_{x_2} \rightarrow x_2 x_3$ for agent x_2 , \dots , $F_{x_{n-1}} X_{n-2} \wedge \neg Exp_{x_{n-1}} \rightarrow x_{n-1} x_n$ for agent x_{n-1} . In other words, an agent wants to call the next agent if he is familiar with the secret of the previous agent and he is not yet an expert.

Note that initially only agent x_1 is active and he will call x_2 , after which both of these agents become active. Due to agent-fairness, agent x_2 will eventually make his call after which agent x_3 becomes active and so on. Note that, for every $i < n$, when the call $x_{i-1} x_i$ becomes active, x_{i-1} already knows all secrets X_1, \dots, X_{i-1} . This means that when eventually the call $x_{n-1} x_n$ takes place, both of the agents involved become the first two experts. Then eventually, the call $x_{n-2} x_{n-1}$ will take place due to agent-fairness, and x_{n-2} will become the third expert and so on until x_1 becomes an expert when the protocol terminates. \square

As pointed out earlier, Theorem 1 implies that the same result holds for rule-fairness. We can also get that every graph with a Hamiltonian path can be a communication graph of an agent-fair / rule-fair correct protocol by simply ignoring the rest of the edges and constructing the protocol as in Theorem 1. But instead we are going to generalize this construction to show this result for any tree, which is any connected acyclic undirected graph. The gossip protocol will be set up in a way to ensure there is a unique directed path from every agent

to a special agent \bar{x} , who is guaranteed to be the first agent to become an expert. Then all the secrets collected in \bar{x} will slowly filter back up the directed paths to all the other agents.

Theorem 2. *For any tree T , there exists an agent-fair correct propositional gossip protocol.*

Proof. Let us pick any agent \bar{x} in this tree T . We now turn this undirected tree into a directed one by setting the direction of all edges in this tree towards \bar{x} . First, the distance of an agent x from \bar{x} is the length of the shortest path from x to \bar{x} traversing just edges of T . Now, formally, the direction of an edge connecting two agents is from the agent at a larger distance from \bar{x} towards the agent at a smaller distance to \bar{x} . This is well-defined, because we now show that no two neighboring agents in T can have the same distance to \bar{x} . If they had then these two paths of equal length, from each of them to \bar{x} , cannot completely overlap. So these paths would intersect at \bar{x} or another agent. The two parts of these paths until the intersection together with the edge connecting these two agents would form a cycle, which is not possible as T was supposed to be acyclic. Similarly, we can argue that each node apart from \bar{x} in such formed tree has a unique outgoing edge (\bar{x} has none). This is because if an agent x has two successors in such a directed tree then there would be a path from each of them to \bar{x} that intersect at some point. These paths to the intersection point together with the edges from x would form a cycle; a contradiction.

Let $pred(x)$ denote the set of all agents who have an outgoing edge to x in such a defined directed tree and $succ(x)$ denote the unique agent who has an incoming edge from x .

Now, the protocol will only consist of a single rule for each agent $x \in G \setminus \{\bar{x}\}$: $\neg Exp_x \wedge \bigwedge_{y \in pred(x)} F_x Y \rightarrow x \circ succ(x)$. In other words, an agent wants to call its unique direct successor agent if he is familiar with all the secrets of his direct predecessors and he is not yet an expert. Agent \bar{x} has no rules.

The rest of the proof proceeds similarly to Theorem 1. First, note that when the call $x \circ succ(x)$ becomes active then x has to already be familiar with all secrets of all his (indirect) predecessors in this directed tree. This is because it can only become active after all direct predecessors of x call him and by inductive assumption all of them have to know all secrets of their (indirect) predecessors at that point. Second, due to agent-fairness, every call that is active will eventually be made, because there is only at most one such call per agent. Initially, only agents that do not have predecessors are active and all of them will make their calls. A call does not stop being active until the agent making it becomes an expert. Due to the protocol setup this cannot happen before \bar{x} becomes an expert, but that only takes place once all the calls that can be made were made (otherwise some secrets would not reach \bar{x}).

Note that once \bar{x} becomes an expert (and the agent calling him), the number of active agents is $n - 2$. Now, only when a new agent becomes an expert (by calling an agent who is already an expert), he stops being active and will never become active again. This shows that the protocol is partially correct. Moreover, as long as there is at least one active call, one of them will have an expert as its callee. This guarantees the protocol to agent-fairly terminate, because eventually such

a call is made and the number of active agents decreases. Together this shows the protocol to be agent-fair correct. \square

We are now ready to prove the result for undirected graphs in its full generality. This result is essentially an almost immediate corollary of Theorem 2.

Corollary 1. *For any undirected connected graph G , there exists an agent-fair and rule-fair correct propositional gossip protocol.*

Proof. Any such graph G has a spanning tree T , which we pick arbitrarily. We use the same protocol as defined in Theorem 2 when applied to T . As such a protocol is agent-fair correct, it is also rule-fair correct. \square

It is trivial to see that for any undirected graph which is not connected there cannot be a correct protocol, because no agent can ever become an expert as secrets cannot be shared across different components. This together with Corollary 1 resolves the case of undirected communication graphs, but naturally leads on to the question of what happens in the directed case.

Theorem 3. *For any strongly connected digraph D , a rule-fair correct propositional gossip protocol exists.*

Proof. The protocol that we are going to use is simply: $\neg Exp_x \rightarrow xy$ for all agents x and agents y which are direct successors of x in D . In other words, an agent keeps on calling every agent he can until he is an expert. Clearly, this protocol can only terminate once every agent is an expert, so it is partially correct.

Now suppose this protocol does not rule-fairly terminate and so there exists a rule-fair computation such that agent a does not learn secret B . As the scheduler is rule-fair, every call which is active will eventually be made. As a is not an expert, and there is a path from a to b , a must be trying to make a call to an agent whose distance to b in D is smaller than that of a . If this agent, say c , already knows B then a would get to know B eventually due to rule-fairness. If c does not yet know B , then c must be trying to make a call to an agent who is closer to b , and so on. As the number of calls in this path that ends at b or at another agent already knowing B is finite, and all these calls are active, eventually all agents along this path will learn B ; a contradiction. \square

Now, we are going to generalize even further the result of Corollary 1 by showing a construction of an agent-fair correct protocol for an arbitrary strongly connected digraph (note that any undirected connected graph can be viewed as a strongly connected digraph).

Theorem 4. *For any strongly connected digraph D , an agent-fair correct propositional gossip protocol exists.*

Proof. Let us number all the agents x_1, \dots, x_n . Intuitively, the constructed protocol will proceed in stages, but agents can progress through stages at a different pace. At stage i an agent tries to learn the secret X_i . Once he achieves that he moves on to stage $i + 1$ until eventually he reaches stage $n + 1$ when he becomes an expert and stops being active. Note that an agent can skip a stage if he already knows the desired secret. It is clear the guard of the rule for an agent

a at stage i can be written down as a propositional formula: $\bigwedge_{j < i} F_a X_j \wedge \neg F_a X_i$.

At the beginning of the protocol, all agents start at stage 1, apart from agent x_1 which can already move on to stage 2. Every agent at stage i tries to make just one call: to any of his direct successors in D whose distance to x_i is the shortest possible in D (breaking ties arbitrarily). Note that this is well-defined, because D is strongly connected.

We now show that at any point of the computation an agent, a , who is at the smallest stage number, i , can reach an agent that knows X_i by following a path created by the calls active at that time. The proof is by induction on the distance of a to x_i . If this distance is 1 then it is obviously true because he can call x_i directly. Otherwise, a either wants to call an agent still at stage i , which by the inductive assumption can reach an agent knowing X_i , or an agent at a higher stage, in which case that agent already knows X_i .

As the scheduler is agent-fair and each agent just tries to make a single call, along the just proven finite path to X_i , all agents will eventually succeed to learn X_i and move to stage $i + 1$. And this process will continue until all agents are experts. \square

Note, however, it is not necessary for the communication graph to be strongly connected. Consider three agents a , b and c , and a directed graph with edges ab and cb only; it is not strongly connected, as there is no directed path between a and c . And yet, an agent-fair correct propositional protocol exists with rules: $\neg Exp_a \rightarrow ab$ for agent a and $\neg Exp_c \rightarrow cb$ for agent c . This protocol simply repeats these two calls until the agent making them is an expert. On the other hand, we cannot even broaden the class of digraphs to weakly connected ones (i.e., connected when the edges' directions are ignored).

Proposition 3. *Not every weakly connected digraph has a rule-fair or agent-fair correct propositional gossip protocol.*

Proof. We prove this with a simple counterexample. Consider three agents a , b and c , and a weakly connected directed graph with edges ba and bc only. W.l.o.g., the computation starts with the call ba followed by bc at a later point. But at that point, both b and c are experts, and hence cannot initiate a call, because otherwise such a call could be repeated forever and the protocol would not terminate. But this means that no further calls may be made, and yet a is still not an expert. \square

4 Complexity of Checking Fair Correctness

We now look at the computational complexity of checking if a given propositional gossip protocol terminates and if it is correct under agent-fair or rule-fair schedulers. With no fairness restrictions on the scheduler used, these problems were shown to be coNP-complete in [Livesey and Wojtczak, 2021b]. We show exactly the same computational complexity but with modified proofs to account for the fairness constraints.

Theorem 5. *Checking if a given propositional gossip protocol agent-fairly or rule-fairly terminates is coNP-complete.*

Proof. Due to monotonicity of the gossip situations along any computation (as agents never forget secrets), the gossip situation can change at most $n^2 - n$ times along any computation.

This is because each of the n agents can know at most n different secrets, and with each change at least one more secret is learned. As the protocol is propositional, for any computation, skipping a call that does not change the current at the time gossip situation will give us another valid computation. This is because the truth value of all propositional formulas used as guards stays the same when that is done and the call sequence can simply proceed with the next call instead. The protocol does not agent-fairly (rule-fairly) terminate iff there is an infinite agent-fair (resp. rule-fair) computation. In such a computation the gossip situation stabilizes after at most $n^2 - n$ changes. Then under agent-fairness, each active agent can then make at least one call that does not change the gossip situation and under rule-fairness all active calls are like that (with at least one agent being active). It is easy to see that we can now guess this polynomial-sized prefix of an infinite computation and which calls should be repeated forever, and verify this guess in polynomial time. This shows that non-termination is in NP and so termination is in coNP.

To show the problem is coNP-hard we will create a polynomial time reduction from the 3-SAT problem, such that termination's NO instances match with 3-SAT's YES instances. The basic idea of this is to have an agent which will learn a certain set of secrets iff the original problem is satisfiable. Only when this agent learns all these secrets, he will be allowed to make a call that can be repeated indefinitely resulting in non-termination of the protocol.

We will call this special agent: the final agent and denote him by f . For each Boolean variable, x , used in the 3-SAT formula, we will create three agents: variable agent (x), true agent (x_{\top}), and false agent (x_{\perp}). For every $i \leq m$, where m is the number of clauses in the 3-SAT formula, we have agent c_i that corresponds to the i -th clause.

Now we define the rules of the protocol for each of the agents. For each variable agent, x , while this agent only knows its own secret X , x wants to call agent x_{\top} or agent x_{\perp} and the scheduler will decide which of these calls takes place. This essentially sets the value of the x variable in the 3-SAT formula to true or false, respectively.

Now any true agent, x_{\top} , and false agent, x_{\perp} , that was called by his corresponding variable agent can make a call to any clause agent, c , whose corresponding clause in the 3-SAT formula uses the literal x or $\neg x$, respectively. This is done with the guard: $X \wedge \neg C$, to ensure this call cannot be repeated.

Once a clause agent receives a call, this means that the corresponding clause is satisfied by the current truth assignment. Once this happens, the clause agent can call f , with a guard: $X \wedge \neg F$, to make a call to f that cannot be repeated.

Now, f will learn the secrets of all clause agents only if all clauses have been satisfied. Agent f can only make a call to some other agent (picked arbitrarily) if he learns all of these clause secrets, with a guard: $C_1 \wedge \dots \wedge C_m$, and so he can repeat this call indefinitely.

The only agent that can repeat a call, and therefore lead to the protocol not terminating is f , and this may only happen if all clauses are satisfied at the same time. Hence, this corresponds directly to YES instances of 3-SAT. If we had a NO instance, it would be impossible for f to learn all clause secrets, and therefore the protocol would terminate as each

call is made at most once then. It is easy to see that this whole construction can be done in polynomial time and space. \square

Theorem 6. *Checking if a given propositional gossip protocol is agent-fair or rule-fair correct is a coNP-complete problem.*

Proof. We show this problem to be in coNP similarly as in Theorem 5 by guessing either a polynomial-sized (i) prefix of an infinite computation or (ii) finite computation after which not all agents end up as experts. It suffices to make a single guess and check whether (i) or (ii) holds in polynomial time.

As for coNP-hardness, the basic set-up is as in Theorem 5 with the following changes. There are three additional agents y, z and l . If an agent $a \in G \setminus \{y, z\}$ does not know L then a can call l . If an agent $a \in G \setminus \{y, z\}$ at any point learns L then a starts to follow the LNS protocol (see Example 2) for the set of agents $G \setminus \{y, z\}$. If an agent $a \in G \setminus \{y, z\}$ knows all secrets apart from Y and Z , he can call y , and if a knows all secrets apart from Z then he can call z .

If y knows all clause secrets as well as F , but not Z , y can call z , and once y also knows Z , he can call f until he becomes an expert. Additionally, if f knows all clause secrets but not L, Y nor Z , he can call y . It is clear that this whole construction can be done in polynomial time and space and all such defined guards are propositional formulas.

Assume that the 3-SAT formula is not satisfiable. Then f cannot learn all clause secrets without learning L , and hence will eventually call (or be called by) an agent which knows L , and hence revert to LNS that excludes y and z . Therefore, y can only be called by an agent which knows all secrets apart from Y and Z . At this stage, agents may begin to call z if they know all secrets apart from Z , thus making both agents experts. Every agent which knows Z is an expert, and so any agent which calls such an agent will also be an expert. We can see that at termination every agent will be an expert.

Assume that the formula is satisfiable. Thus it is possible to be in a situation where f knows all clause secrets but not L, Y nor Z , so can call y . After this call, y may call z , and then repeatedly call f , until y becomes an expert (which he is not yet as he does not know L). Now, after the first instance of the yf call, every agent which knows Y also knows Z . This means that an agent either knows both Y and Z , or neither Y nor Z . As z can never make a call, this means z must be called by an agent which knows L in order to learn this secret. Yet, for an agent to call z , he must first learn Y . Now, every agent which knows Y also knows Z , and so no agent that knows L will ever call z , hence z will never become an expert. \square

5 Conclusions

In this paper we have revisited the solution to Problem 5 of [Apt and Wojtczak, 2019] given in [Livesey and Wojtczak, 2021b]. We showed that the type of the scheduler used makes a huge difference to the answer to this problem: while unrestricted schedulers force the communication graph to be complete, any strongly connected digraph admits an agent-fair and rule-fair correct propositional gossip protocols. Pinning down precisely for which class of digraphs this property holds is still open. We also showed agent-fair and rule-fair termination and correctness of such protocols to be coNP-complete.

Acknowledgements

We would like to thank the anonymous reviewers whose comments helped to improve this paper. The first author was supported by EPSRC NPIF PhD studentship.

References

- [Apt and Wojtczak, 2017] Krzysztof R. Apt and Dominik Wojtczak. Decidability of fair termination of gossip protocols. In *Proc. of the 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR 21)*, pages 73–85, 2017.
- [Apt and Wojtczak, 2018] Krzysztof R. Apt and Dominik Wojtczak. Verification of distributed epistemic gossip protocols. *J. Artif. Intell. Res. (JAIR)*, 62:101–132, 2018.
- [Apt and Wojtczak, 2019] Krzysztof R. Apt and Dominik Wojtczak. Open problems in a logic of gossips. In *Proceedings Seventeenth Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2019)*, volume 297 of *EPTCS*, pages 1–18, 2019.
- [Apt et al., 2016] Krzysztof R. Apt, Davide Grossi, and Wiebe van der Hoek. Epistemic protocols for distributed gossiping. In *Proceedings of the 15th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2015)*, volume 215 of *EPTCS*, pages 51–66, 2016.
- [Apt et al., 2017] Krzysztof R. Apt, Eryk Kopczyński, and Dominik Wojtczak. On the computational complexity of gossip protocols. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*, pages 765–771, 2017.
- [Apt et al., 2018] Krzysztof R. Apt, Davide Grossi, and Wiebe van der Hoek. When are two gossips the same? In Gilles Barthe, Geoff Sutcliffe, and Margus Veanes, editors, *LPAR-22. 22nd International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, pages 36–55. EasyChair, 2018.
- [Attamah et al., 2014] Maduka Attamah, Hans Van Ditmarsch, Davide Grossi, and Wiebe van der Hoek. Knowledge and gossip. In *Proceedings of ECAI’14*, pages 21–26. IOS Press, 2014.
- [Bumby, 1981] Richard T Bumby. A problem with telephones. *SIAM Journal on Algebraic Discrete Methods*, 2(1):13–18, 1981.
- [Cooper et al., 2016] Martin C. Cooper, Andreas Herzig, Faustine Maffre, Frédéric Maris, and Pierre Régnier. Simple Epistemic Planning: Generalised Gossiping. In *Proceedings of ECAI 2016*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 1563–1564. IOS Press, 2016.
- [Cooper et al., 2018] Martin C. Cooper, Andreas Herzig, Frédéric Maris, and Julien Vianey. Temporal epistemic gossip problems. In *European Conference on Multi-Agent Systems*, pages 1–14. Springer, 2018.
- [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.
- [Harary and Schwenk, 1974] Frank Harary and Allen J Schwenk. The communication problem on graphs and digraphs. *Journal of the Franklin Institute*, 297(6):491–495, 1974.
- [Hedetniemi et al., 1988] Sandra M. 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.
- [Hromkovič et al., 2005] Juraj Hromkovič, 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 *Proc. of the 44th Annual IEEE Symposium on Foundations of Computer Science, FOCS ’03*, 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.
- [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.
- [Livesey and Wojtczak, 2021a] Joseph Livesey and Dominik Wojtczak. Minimal number of calls in propositional protocols. In *International Conference on Reachability Problems*, pages 132–148. Springer, 2021.
- [Livesey and Wojtczak, 2021b] Joseph Livesey and Dominik Wojtczak. Propositional gossip protocols. In *Proc. of 23rd International Symposium on Fundamentals of Computation Theory (FCT 2021)*, pages 354–370. Springer, 2021.
- [Tijdeman, 1971] Robert Tijdeman. On a telephone problem. *Nieuw Archief voor Wiskunde*, 3(XIX):188–192, 1971.
- [van Ditmarsch and Kokkinis, 2017] Hans van Ditmarsch and Ioannis Kokkinis. The expected duration of sequential gossiping. In *Multi-Agent Systems and Agreement Technologies*, pages 131–146. Springer, 2017.
- [van Ditmarsch et al., 2017a] Hans van Ditmarsch, Ioannis Kokkinis, and Anders Stockmarr. Reachability and expectation in gossiping. In *PRIMA 2017: Principles and Practice of Multi-Agent Systems*, pages 93–109. Cham, 2017. Springer International Publishing.
- [van Ditmarsch et al., 2017b] Hans van Ditmarsch, Jan van Eijck, Pere Pardo, Rahim Ramezani, and François Schwarzentruber. Epistemic protocols for dynamic gossip. *J. of Applied Logic*, 20(C):1–31, 2017.
- [van Ditmarsch et al., 2020] Hans van Ditmarsch, Wiebe van Der Hoek, and Louwe B Kuijer. The logic of gossiping. *Artificial Intelligence*, 286:103306, 2020.