Distributed Optimal Allocation with Quantized Communication and Privacy-Preserving Guarantees

Jakob Nylöf* Apostolos I. Rikos* Sebin Gracy** and Karl Henrik Johansson*

* Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, and Digital Futures, Stockholm, Sweden. jnylof@kth.se, rikos@kth.se, kallej@kth.se
** Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA. sebin.gracy@rice.edu

Abstract: In this paper, we analyze the problem of optimally allocating resources in a distributed and privacy-preserving manner. We propose a novel distributed optimal resource allocation algorithm with privacy-preserving guarantees, which operates over a directed communication network. Our algorithm converges in finite time and allows each node to process and transmit quantized messages. Our algorithm utilizes a distributed quantized average consensus strategy combined with a privacy-preserving mechanism. We show that the algorithm converges in finite-time, and we prove that, under specific conditions on the network topology, nodes are able to preserve the privacy of their initial state. Finally, to illustrate the results, we consider an example where test kits need to be optimally allocated proportionally to the number of infections in a region. It is shown that the proposed privacy-preserving resource allocation algorithm performs well with an appropriate convergence rate under privacy guarantees.

Keywords: Distributed Algorithms, Optimal Resource Allocation, Privacy-Preservation, Distributed Optimization

1. INTRODUCTION

In distributed systems and networks, various components (nodes) are often required to allocate a set of resources in an optimal way such that specific performance objectives are satisfied. Distributed optimal resource allocation is an optimization problem, and has many applications such as optimally scheduling tasks for data centers (Rikos et al., 2021c), optimally coordinating the response of a set of distributed energy resources (Zholbarysov et al., 2020), optimally allocating vaccines/tests for pandemic stabilization (Ma et al., 2021). Note that in the current literature there exist a variety of centralized algorithms for addressing optimal resource allocation problems (see for instance (Fang, 2013; Lotfi et al., 2012)). However, a central entity could possibly suffer from processing issues due to network scale, and may also impose privacy risks (due to the gathering of all available data to a central entity). For these reasons, we aim to address the optimal resource allocation problem in a distributed fashion.

Distributed optimization algorithms have received great attention recently, due to the wide variety of applications which range from distributed estimation to machine learning (Nedić et al., 2018; Yang et al., 2019). However, a vast majority of algorithms in the current literature assume that the messages exchanged among nodes consist of real values with infinite precision (see, for instance, (Grammenos et al., 2020; Preciado et al., 2014; Ramírez-Llanos and Martínez, 2018; Beck et al., 2014)) and they exhibit asymptotic convergence within some error (see (Domínguez-García and Hadjicostis, 2015)). Furthermore, most algorithms typically do not provide privacy-preserving guarantees (see (Zholbarysov et al., 2020; Rikos et al., 2021b)). In this paper, we aim to address both of these issues, since in the current literature there is a need for finite-time distributed optimal resource allocation algorithms with privacy-preserving guarantees and efficient communication, which exhibit finite time convergence. To illustrate the efficacy of our proposed algorithm, we consider the setting where vaccines (i.e., devices for testing whether a person is infected from a specific virus) have to be distributed in an optimal fashion over a network of cities dealing with an epidemic outbreak.

Main Contributions. Our main contributions are as follows.

• We present an optimal allocation algorithm with quantized communication and privacy-preservation guarantees; see Algorithm 1. Furthermore, during the operation of our algorithm, each node terminates its operation once convergence has been achieved. Note that it is the first distributed stopping mechanism adjusted to the algorithm’s necessary privacy-
preservation guarantees. Our algorithm’s operation is applied to distributed optimal test kit allocation problem over strongly connected networks.

- We analyze the convergence of Algorithm 1, and we show that all nodes calculate the optimal allocation in finite time with high probability; see Theorem 4.
- We provide sufficient topological conditions for privacy-preservation of Algorithm 1; see Theorem 5.

The optimal allocation algorithm in this paper uses properties of quantized average consensus algorithms (Rikos et al., 2021d; Aysal et al., 2008; Amini et al., 2019; Zhang and Liu, 2020; Lavacchi and Murray, 2012; Kashyap et al., 2007; El Chamie et al., 2016) that allow nodes to exchange quantized messages. Transmissions of quantized offsets is performed in a deterministic manner. In contrast, the randomized nature of the quantized average consensus algorithm, as the injection of quantized offsets is performed according to a framework in order to formulate the problem of interest (Rikos et al., 2021b; Wang, 2019; Kefayati et al., 2007; Manitara and Hadjicostis, 2013; Gupta et al., 2017). In particular, (Rikos et al., 2021b) utilizes the injection of random quantized offsets into interaction messages transmitted from private nodes. However, the injection of quantized offsets is done in a deterministic manner. In contrast, in our paper the privacy preserving strategy is adjusted to the randomized nature of the quantized average consensus algorithm, as the injection of quantized offsets is performed according to a set of event-triggered conditions.

2. NOTATION AND BACKGROUND

The sets of real numbers, positive real numbers, integers and natural numbers are denoted by \( \mathbb{R} \), \( \mathbb{R}_+ \), \( \mathbb{Z} \) and \( \mathbb{N} \), respectively. For any \( a \in \mathbb{R} \), the floor is defined as \( \lfloor a \rfloor = \sup \{ b \in \mathbb{Z} \mid b \leq a \} \) and the ceiling as \( \lceil a \rceil = \inf \{ b \in \mathbb{Z} \mid b \geq a \} \).

Graph-Theoretic Notions. The communication network is represented by a strongly connected directed graph (digraph) \( G_d = (V, E) \) of \( n \) nodes. In digraph \( G_d \), \( V = \{ v_1, v_2, \ldots, v_n \} \) is the set of nodes, whose cardinality is denoted as \( n = |V| \geq 2 \), and \( E \subseteq V \times V - \{ (v_j, v_j) \mid v_j \in V \} \) is the set of edges (self-edges excluded) whose cardinality is denoted as \( m = |E| \). We assume that the given digraph \( G_d = (V, E) \) is strongly connected (i.e., for each pair of nodes \( v_j, v_i \in V, v_i \neq v_j \), there exists a directed path from \( v_i \) to \( v_j \)). The diameter \( D \) of a digraph is the longest shortest path between any two nodes \( v_j, v_i \in V \) in the network. The set of in-neighbors of \( v_j \) is represented by \( N_j^- = \{ v_i \in V \mid (v_j, v_i) \in E \} \), and it is the subset of nodes that can directly transmit information to node \( v_j \) is called the in-degree of \( v_j \) and is denoted by \( D_j^- = |N_j^-| \). The set of out-neighbors of \( v_j \) is represented by \( N_j^+ = \{ v_i \in V \mid (v_i, v_j) \in E \} \), and it is the subset of nodes that can directly receive information from node \( v_j \). The out-degree of \( v_j \) is denoted by \( D_j^+ = |N_j^+| \).

3. PROBLEM FORMULATION

3.1 Distributed Optimal Resource Allocation Problem

We state the following optimization problem, which is inspired by (Rikos et al., 2021c). For each node \( v_j \in V \), we define the scalar quadratic local cost function \( f_j : \mathbb{R} \rightarrow \mathbb{R} \) as

\[
f_j(z) = \frac{1}{2} \alpha_j (z - \chi_j)^2,
\]

where \( \alpha_j \in \mathbb{R}_+ \), \( \chi_j \in \mathbb{R}_+ \) is the demand at node \( v_j \), and \( z \) the global optimization parameter. In (1) we capture the cost of the node \( v_j \) agreeing to obtain the quantity \( z \) in relation to its demand \( \chi_j \), where the weight \( \alpha_j \) scales the cost.

The global cost function is the sum of the local cost functions (1) corresponding to each node \( v_j \in V \). The global cost function is the total cost of all nodes in the network agreeing to obtain the parameter \( z \). Consequently, each node \( v_j \) aims to obtain a value \( z^* \) which minimizes the global cost function

\[
z^* = \arg \min_{z \in \mathbb{R}} \sum_{v_i \in V} f_i(z),
\]

Equation (2) has a closed form solution given by

\[
z^* = \frac{\sum_{v_i \in V} \alpha_i \chi_i}{\sum_{v_i \in V} \alpha_i}.
\]

Note that if \( \alpha_i = 1 \) for all \( v_i \in V \), then the solution is the average of the initial states.

3.2 Modification of Optimal Resource Allocation Problem

Consider an optimization step \( m \) which represents a day on which we aim to find an optimal allocation of test kits to number of infections. For every node \( v_j \in V \), denote the local number of stored test kits by \( u_j[m] \), received test kits by \( l_j[m] \), and number of infections by \( \lambda_j[m] \). Note here that these quantities are positive integers (which enables efficient communication since they are quantized values). Define \( w_j^*[m] \) as the number of test kits added (or, if negative, subtracted) to the stored test kits in order to achieve the optimal allocation of the available test kits. We refer to \( w_j^*[m] \) as the optimal allocation. Furthermore, denote the global number of stored test kits by \( u_{tot}[m] = \sum_{v_i \in V} u_i[m] \), global number of received test kits by \( l_{tot}[m] = \sum_{v_i \in V} l_i[m] \) and global number of infections by \( \lambda_{tot}[m] = \sum_{v_i \in V} \lambda_i[m] \). We drop the index \( m \) in the sequel (since we aim to find the optimal allocation of test kits in the same way during each optimization step).

We now state following problem \textbf{P1}. which will be used as a framework in order to formulate the problem of interest in this paper (defined as Problem 1 at the end of this section).

\textbf{P1.} Formulate a distributed algorithm that allows each node \( v_j \) to calculate the optimal allocation \( w_j^* \) so that its local ratio of test kits to number of infections equals the global ratio of test kits to number of infections in the entire network.

To solve \textbf{P1.}, we aim to find \( w_j^* \) such that...
Note that $q = (\sum_{v_i \in V} \lambda_i (l_i + w)) / (\sum_{v_i \in V} \lambda_i)$ is the same as (3) with $\alpha_j = \lambda_j$, and $\chi_j = (l_j + u_j) / \lambda_j$ for all $v_j \in V$. Equation (4) thus implies that $(w^*_j + u_j) / \lambda_j$ is the solution to the optimization problem (2) where the weight $\alpha_j$ is the number of infections and $\lambda_j$ the initial test kits to number of infections located at every node. Hence, we require every node to calculate the global test kits to number of infections given by (5) and then solve for $w^*_j$ in (4).

The quantized coordination algorithm considered in this paper (Rikos et al., 2021d) allows each node to calculate either the ceiling or the floor of $q$ which yields the optimal allocation

$$w^*_j = \lceil q \lambda_j - u_j \rceil \text{ or } \lfloor q \lambda_j - u_j \rfloor, \forall v_j \in V.$$  

Equation (6) may introduce a larger quantization error compared to solving for $w^*_j$ in (4). However, the event-triggering operation and the exchange of integer-valued messages increases the efficiency of communication while it maintains a fast convergence speed.

### 3.3 Distributed Privacy-Preserving Optimal Resource Allocation Problem

The problem we present in this paper is denoted as Problem 1. It is borrowed from (Rikos et al., 2021b) and it is adjusted to the optimal allocation scenario. Consider a strongly connected digraph $G = (V, E)$, where $|V| \geq 3$. The node set $V$ is partitioned into three subsets: i) a subset of nodes $v_j \in V_p \subset V$ that wish to preserve their privacy by not revealing their initial states to other nodes, ii) the subset of nodes $v_i \in V_c \subset V$ that are curious (i.e., they try to identify the initial states of all or a subset of nodes in the network and they are possibly colluding among themselves), and iii) the subset of nodes $v_p \in V_n \subset V$ that are neutral (i.e., they neither wish to preserve their privacy nor identify the initial states of other nodes). An example is shown in Fig. 1 (borrowed from (Rikos et al., 2021b)).

![Fig. 1. Example of a digraph with the different types of nodes in the network: nodes $v_1, v_2 \in V_p$ that wish to preserve their privacy, node $v_3 \in V_c$ that is curious (wishes to identify the initial states of other nodes), and node $v_4 \in V_n$ that is neither curious nor wishes to preserve its privacy.](image)

We now define the problem of interest in our paper.

**Problem 1.** In our paper we aim to develop a distributed optimal allocation algorithm for nodes $v_j \in V_p$ that wish to preserve their privacy when they exchange quantized information with neighboring nodes while calculating $w^*_j$ which fulfills (4). Furthermore, nodes need to (i) converge to the optimal solution after a finite number of time steps, (ii) process and transmit quantized values, and (iii) cease transmissions once convergence has been achieved.

### 4. DISTRIBUTED TEST KIT ALLOCATION WITH PRIVACY-PRESERVATION

#### 4.1 Quantized Privacy-Preserving Strategy

During the operation of our algorithm we aim to calculate $w^*_j$ which fulfills (4) while preserving the privacy of the nodes following the privacy preserving strategy. In the current literature (e.g., Kefayati et al., 2007; Manzoor and Hadjicostis, 2013; Mo and Murray, 2017; Charalambous et al., 2019) and references therein), each node initially injects a nonzero offset to its initial state. During the operation of our algorithm, if a node follows the proposed privacy preserving strategy it assigns an offset to each outgoing link (note that the sum of offsets is equal to its initial state). Then, if it performs a transmission towards an out-neighbor, it injects the assigned offset to the transmitted messages. More specifically, each node $v_j \in V_p$ maintains a set of values $\text{off}^{(z)} = 0 \neq \text{off}^{(y)} \neq 0$, for every $v_i \in N^+_j$. The sum of these values is equal to the node’s initial state (i.e., $\sum_{v_i \in N^+_j} \text{off}^{(z)} = l_j + u_j$, and $\sum_{v_i \in N^+_j} \text{off}^{(y)} = \lambda_j$).

Furthermore, each node $v_j \in V_p$ maintains a set of counters $\text{off}^{(z)} = 1$, for every $v_i \in N^+_j$ in order to remember whether it has injected every offset to the transmitted messages. Then, if node $v_j$ performs a transmission towards out-neighbor $v_i \in N^+_j$, it injects $\text{off}^{(z)}$ and $\text{off}^{(y)}$ to the transmitted messages $z_j$, $y_j$ respectively. Finally, note that the nodes $v_i \notin V_p$, either execute the proposed algorithm or execute the quantized average consensus algorithm in (Rikos et al., 2021a).

#### 4.2 Optimal Allocation Algorithm with Privacy-Preserving Guarantees

We now present the distributed algorithm (detailed below as Algorithm 1) which solves Problem 1 described in Section 3.3. In order to solve Problem 1 we need to consider the following assumptions.

**Assumption 2.** The communication network is modelled as a strongly connected digraph.

**Assumption 3.** An upper bound $D'$ of the diameter $D$ (i.e., $D' \geq D$) is known to every node in the network.

Assumption 2 ensures that information transmitted by one node can reach every other node, and is important for guaranteeing convergence to the optimal solution. Assumption 3, is required for terminating the operation of Algorithm 1 once convergence has been achieved.
Algorithm 1. Quantized Test Kit Allocation Algorithm With Privacy-Preservation

**Input:** A strongly connected digraph $G_d = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ edges. Each node $v_i \in \mathcal{V}$ has knowledge of $l_i, u_i, D, \lambda_j \in \mathbb{Z}$.

**Initialization:** Each node $v_i \in \mathcal{V}$ does the following:

1. Assigns a nonzero probability $b_{ij}$ to each of its outgoing edges $m_{ij}$, where $v_i \in N_j^+ \cup \{v_j\}$, as follows:
   \[
   b_{ij} = \begin{cases} 
   \frac{1}{1 + D_j^+} & \text{if } l = j \text{ or } v_i \in N_j^+, \\
   0 & \text{if } l \neq j \text{ and } v_i \notin N_j^+.
   \end{cases}
   \]

2. Sets $z_j[0] := \lambda_j, y_j[0] = t_j + u_j$, and $\text{flag}_{ij} = 0$.

3. Sets $\text{off}_{ij}^{(z)} \neq 0$, for every $v_i \in N_j^+$, such that $\sum_{v_i \in N_j^+} \text{off}_{ij}^{(z)} = t_j + u_j$, and $\text{off}_{ij}^{(y)} \neq 0$, for every $v_i \in N_j^+$, such that $\sum_{v_i \notin N_j^+} \text{off}_{ij}^{(y)} = \lambda_j$.

4. Sets $\text{off}_{ij} = 1$, for every $v_i \in N_j^+$.

**Iteration:** For $k = 1, 2, \ldots, \ldots$, each node $v_i \in \mathcal{V}$, does the following:

- **while** $\text{flag}_{ij} = 0$ **do**
  - if $k \text{ mod } D = 1$ then sets $M_j = [y_j[k]/z_j[k]], m_j = [y_j[k]/z_j[k]]$;
  - if $\sum_{v_i \in N_j^+} \text{off}_{ij}^{(z)} > 0$ then sets $M_j = M_j + 2$;
  - broadcasts $M_j, m_j$ to every $v_i \in N_j^+$;
  - receives $m_i$, from every $v_i \in N_j^+$;
  - sets $M_j = \max_{v_i \in N_j^+ \cup \{v_i\}} M_i, m_j = \min_{v_i \in N_j^+ \cup \{v_i\}} m_i$;
  - if $z_j[k] > 1$, then calls Algorithm 1A;
  - if $z_j[k] \leq 1$, sets $c_{ij}^{(y)}[k] = y_j[k], c_{ij}^{(z)}[k] = z_j[k]$;
  - receives $c_{ij}^{(y)}[k], c_{ij}^{(z)}[k]$ from every $v_i \in N_j^+$ and sets
    \[
    y_j[k + 1] = c_{ij}^{(y)}[k] + \sum_{v_i \in N_j^+} w_{ji}[k] y_j[k],
    \]
    \[
    z_j[k + 1] = c_{ij}^{(z)}[k] + \sum_{v_i \in N_j^+} w_{ji}[k] z_j[k],
    \]

- where $w_{ji}[k] = 1$ if node $v_j$ receives $c_{ij}^{(y)}[k], c_{ij}^{(z)}[k]$ from $v_i \in N_j^+$ at iteration $k$ (otherwise $w_{ji}[k] = 0$);
  - if $k \text{ mod } D = 0$ then, if $M_j - m_j \leq 1$ then sets $y_j[k] = [\lambda_j q_j[k]]$ and $\text{flag}_{ij} = 1$.

**Output:** (4) is fulfilled for every $v_i \in \mathcal{V}$.

The intuition behind Algorithm 1 is the following. Initially, each node in the set $\mathcal{V}_p$ calculates a set of offsets; one offset for each out-neighbor. Then, each node executes the quantized average consensus algorithm in (Rikos et al., 2021a). During the operation of the quantized average consensus algorithm, if one node in the set $\mathcal{V}_p$ performs a transmission towards an out-neighbor, it injects the calculated offset to the transmitted variables. Finally, if one node in the set $\mathcal{V}_p$ has not transmitted every offset to each out-neighbor, it delays the distributed stopping protocol until every offset is transmitted.

Note here that every node in the set $\mathcal{V}_p$ that wants to preserve its privacy executes Algorithm 1. The set of neutral nodes in $\mathcal{V}_n$ executes the quantized average consensus algorithm in (Rikos et al., 2021a). Finally, the set of curious nodes in $\mathcal{V}_c$, either executes Algorithm 1 or the quantized average consensus algorithm in (Rikos et al., 2021a) (this means that $\mathcal{V}_p$ and $\mathcal{V}_c$ are not necessarily disjoint).

Next, we show that Algorithm 1 solves Problem 1 in Section 3.3. Due to space limitations we provide a sketch of the proof.

**Theorem 4.** Consider a strongly connected digraph $G_d = (\mathcal{V}, \mathcal{E})$ under Assumptions 2, 3. Every node in the set (i) $\mathcal{V}_p$ executes Algorithm 1, (ii) $\mathcal{V}_n$ executes the algorithm in (Rikos et al., 2021a), and (iii) $\mathcal{V}_c$ either executes Algorithm 1 or the algorithm in (Rikos et al., 2021a). Algorithm 1 solves Problem 1.

**Proof:** The main idea of this proof is that we will calculate (i) the number of time steps in order for every $v_i \in \mathcal{V}_p$ to complete the privacy preservation mechanism (i.e., to inject all its offsets in the network), and (ii) the number of time steps for the algorithm in (Rikos et al., 2021a) to converge.

The operation of Algorithm 1 can be interpreted as the “random walk” of $2\lambda_{tot} - n$ “tokens” in a Markov chain (where $\lambda_{tot} = \sum_{v_i \in \mathcal{V}} \lambda_j$ and $n = |\mathcal{V}|$). Furthermore, every node has one stored token which is stationary (i.e., it does not perform a random walk). Each token contains a pair of values $y[k] \in \mathbb{N}, z[k] = 1$. Each time two or more tokens meet at a specific node, their $y[k]$ values either become equal or have difference equal to one.

From (Rikos et al., 2021c, Lemma 1) we have that the probability $P_{T_{tot}}^{D+1}$ that “the specific token $T_{out,\theta}$ is at node $v_i$ after $D+1$ time steps, and node $v_i$ transmits to a specific $v_i' \in N_j^+$ is

\[
P_{T_{tot}}^{D+1} \geq (1 + D_{max})^{-(D+1)}.
\]
This means that the probability \( P_{\mathcal{N},T_\text{out},\theta}^{D+1} \) that “the specific token \( T_\text{out},\theta \) has not visited node \( v_i \) after \( D+1 \) time steps (or has visited but not been transmitted to the specific node \( v_i \in \mathcal{N}_i^+ \)) is

\[
P_{\mathcal{N},T_\text{out},\theta}^{D+1} \leq 1 - (1 + D_{\max}^+)^{(D+1)},
\]

(10)

By extending this analysis, we can state that for any \( \epsilon \), where \( 0 < \epsilon < 1 \) and after \( \tau(D+1) \) time steps where

\[
\tau \geq \left[ \log \epsilon \log (1 - (1 + D_{\max}^+)^{(D+1)}) \right],
\]

(11)

the probability \( P_{\mathcal{N},T_\text{out},\theta}^{\tau} \), that “the specific token \( T_\text{out},\theta \) has not visited node \( v_i \) after \( \tau(D+1) \) time steps (or has visited but not been transmitted to the specific node \( v_i \in \mathcal{N}_i^+ \)) is

\[
P_{\mathcal{N},T_\text{out},\theta}^{\tau} \leq [P_{\mathcal{N},T_\text{out},\theta}^{D+1}]^\tau \leq \epsilon.
\]

(12)

This means that after \( \tau(D+1) \) time steps, where \( \tau \) fulfills (11), the probability that “the specific token \( T_\text{out},\theta \) has visited node \( v_i \) after \( \tau(D+1) \) time steps and has been transmitted to a specific \( v_i_i \in \mathcal{N}_i^+ \) is equal to \( 1 - \epsilon \).

Thus, by extending this analysis, for \( k \geq (D_{\max}^+)\tau(D+1) \) we have that every node \( v_i \) will perform a transmission towards every out-neighbor \( v_i_i \in \mathcal{N}_i^+ \) with probability \((1 - \epsilon)^{(D_{\max}^+)}\).

Once every node \( v_i \) performs a transmission towards every out-neighbor \( v_i_i \in \mathcal{N}_i^+ \), the privacy preserving strategy has been completed, and the operation of Algorithm 1 is similar to (Rikos et al., 2021c). As a result, for the operation of Algorithm 1 during time steps \( k \geq (D_{\max}^+)\tau(D+1) \) the rest of the proof is similar to Theorem 3 in (Rikos et al., 2021c) (since the operation of Algorithm 1 for time steps \( k \geq (D_{\max}^+)\tau(D+1) \) is identical to (Rikos et al., 2021c)).

4.3 Topological Conditions for Privacy Preservation

We now present, in the following theorem, the necessary topological conditions for privacy preservation.

**Theorem 5.** Consider a fixed strongly connected digraph \( \mathcal{G}_d = (\mathcal{V}, \mathcal{E}) \) under Assumptions 2, 3. Every node in the set \( (i) \mathcal{V}_p \) executes Algorithm 1, (ii) \( \mathcal{V}_s \) executes the algorithm in (Rikos et al., 2021a), and (iii) \( \mathcal{V}_c \), either executes Algorithm 1 or the algorithm in (Rikos et al., 2021a). No subset of curious nodes \( \mathcal{V}_c \) is able to identify the initial state \( y_j[0] \) of \( v_j \), if, and only if, the following conditions are fulfilled:

i) \( v_j \) has at least one out-neighbor (or in-neighbor) \( v_j \in \mathcal{V}_p \setminus (\mathcal{V}_s \cup \{v_j\}) \),

ii) there is a message exchange between \( v_j \) and \( v_i \) while both are implementing the privacy-preserving mechanism,

iii) \( v_j \) transmits to an out-neighbor \( v_i \) for the first time during the next time step.

**Proof:** The proof consists of two parts. In the first part, we analyze the sufficiency of the above conditions (i) - (iii), and in the second part we analyze their necessity.

Regarding the first part, let us assume that conditions (i) - (iii) hold. Let us assume that nodes \( v_j \) and \( v_i \) are executing Algorithm 1 (i.e., \( v_j \in \mathcal{V}_p \)). Now let us assume that at time step \( k' \), node \( v_j \) transmits a message to its out-neighbor \( v_i \) (the case \( v_i \in \mathcal{N}_j^- \) can be proven identically).

Node \( v_j \) will inject \( \text{off}^{(y)}_{lj} \), \( \text{off}^{(z)}_{lj} \) to the transmitted values. The values \( \text{off}^{(y)}_{lj} \), \( \text{off}^{(z)}_{lj} \) are only known to \( v_j \) and perhaps to \( v_l \). Then, in the next time step, node \( v_l \) will transmit to \( v_i \) an out-neighbor \( v_i \) for the first time. This means that \( v_i \) will inject \( \text{off}^{(y)}_{lj} \), \( \text{off}^{(z)}_{lj} \) to the transmitted values. As a result, the transmitted message depends on the sum of offsets \( \text{off}^{(y)}_{lj} + \text{off}^{(z)}_{lj} \) and \( \text{off}^{(y)}_{lj} + \text{off}^{(z)}_{lj} \). Since, both \( v_j, v_l \in \mathcal{V}_p \), the curious nodes may be able to determine \( \text{off}^{(y)}_{lj} + \text{off}^{(z)}_{lj} \), and \( \text{off}^{(y)}_{lj} + \text{off}^{(z)}_{lj} \), but not each \( \text{off}^{(y)}_{lj} \), \( \text{off}^{(z)}_{lj} \), and \( \text{off}^{(y)}_{lj}, \text{off}^{(z)}_{lj} \). As a result, the privacy of both node \( v_j \) and node \( v_i \) is preserved.

Regarding the second part, let us assume that condition (i) does not hold. In this case, all the in- and out-neighbors of node \( v_j \) are curious and they collude with each other. This means that the curious nodes will know all the values node \( v_j \) transmitted to its out-neighbors, and they will know all the values \( v_j \) received from its in-neighbors. Therefore, it is not possible for node \( v_j \) to keep its privacy. Let us now assume that condition (ii) does not hold. In this case, non of the in- or out-neighbors of node \( v_j \) will inject any offsets to the messages they transmit. This means that the curious nodes will know that the transmitted values have only the injected offsets from node \( v_j \). Therefore, it is not possible for node \( v_j \) to keep its privacy. Finally, the case where condition (iii) does not hold, the claim can be proven analogously.

5. SIMULATION RESULTS

![Fig. 2. Convergence](image)

Calculation of the optimal allocation of tests

Calculation of the global ratio of tests to cases

We now illustrate the efficiency of Algorithm 1. To this end, we consider the setting where test kits need to
corresponds to the calculated value of the global ratio $\lambda$ variable in Fig. 2b). In Fig. 2a each line represents the state $q_j$ ceiling or the floor of $u_j$ in (5) for every iteration step. In Fig. 2a we have that all state variables have converged to either the ceiling or the floor of $u_j$. Each line is represented by $w_j^* + u_j = q_j^*[k] \lambda_j$ at every iteration step. In Fig. 2b, the privacy preservation mechanism can be seen as “spikes” extending from the lines of Figure 2. These “spikes” denote the offset injection during Iteration Step 5 of Algorithm 1A.

In Figure 3 we show the mean number of iterations required for convergence of Algorithm 1 without and with privacy preservation guarantees (i.e., if we execute Algorithm 1 or the quantized average consensus algorithm in (Rikos et al., 2021a)). We consider networks of 10 and 100 nodes, and assume that these are strongly connected networks. We implement Algorithm 1 for both these networks; see Fig. 3a and Fig. 3b, respectively. The number of test kits $t_j + u_j$ is randomly set in the interval $[500, 1500]$ at each node. In Fig. 3a, both algorithms (i.e., with and without the privacy preservation mechanism) require the same number of time steps for convergence. The same holds for Fig. 3b for $\lambda_j$ greater than 15 infections. However, in Fig. 3b, we have that Algorithm 1 requires more time steps for convergence when $\lambda_j < \{1, 2, ..., 10\}$.

6. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we presented a novel distributed privacy-preserving algorithm that optimally allocates test kits proportionally to the number of infections. We showed that all nodes calculate the optimal allocation of test kits with high probability after a finite number of time steps while exchanging quantized values. Furthermore, once convergence has been achieved every node terminates its operation. We also provided sufficient topological conditions for privacy-preservation. Finally we presented simulation results of our proposed distributed algorithm, and we demonstrated its convergence rate for networks of various sizes.

In the future, we plan to extend our algorithm to also handle errors in the interaction messages transmitted between nodes.

REFERENCES

Amini, A., Asif, A., and Mohammadi, A. (2019). Quantized event-triggered sampled-data average consensus with guaranteed rate of convergence. In ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4614–4618. doi:10.1109/ICASSP.2019.8682877.

Aysal, T.C., Coates, M.J., and Rabbat, M.G. (2008). Distributed average consensus with dithered quantization. IEEE Transactions on Signal Processing, 56(10), 4905–4918. doi:10.1109/TSP.2008.927071.

Beck, A., Nedić, A., Ozdaglar, A., and Teboulle, M. (2014). An $o(1/k)$ gradient method for network resource allocation problems. IEEE Transactions on Control of Network Systems, 1(1), 64–73. doi: 10.1109/TCNS.2014.2309751.

Charalambous, T., Manitara, N.E., and Hadjicostis, C.N. (2019). Privacy-preserving average consensus over digraphs in the presence of time delays. Allerton Conference on Communication, Control, and Computing, 238–245.

Domínguez-García, A.D. and Hadjicostis, C.N. (2015). Distributed resource coordination in networked systems described by digraphs. Systems & Control Letters, 82, 33–39. doi: https://doi.org/10.1016/j.sysconle.2015.04.012.

El Chamie, M., Liu, J., and Ba¸sar, T. (2016). Design and analysis of distributed averaging with quantized communication. IEEE Transactions on Automatic Control, 61(12), 3870–3884. doi:10.1109/TAC.2016.2530939.

Fang, L. (2013). A generalized dea model for centralized resource allocation. European Journal of Operational Research, 228(2), 405–412.
