Distributed Finite-Time Privacy-Preserving Optimal Allocation of Test Kits for Controlling Pandemic Spreading

Jakob Nylòf, Apostolos I. Rikos, Sebin Gracy, and Karl H. Johansson.

Abstract—In this paper, we analyze the problem of optimally allocating resources in a distributed and privacy-preserving manner. We focus on the scenario where different cities/entities over a country aim to optimally allocate test kits according to their number of infections for controlling the spread of a pandemic. We propose a distributed privacy-preserving optimal resource allocation algorithm with efficient (i.e., quantized) communication over a directed communication network. The proposed algorithm allows each node to calculate the optimal allocation of test kits so that each node's ratio of test kits to infections equals the global ratio of test kits to infections in the network. The algorithm utilizes a fast distributed probabilistic coordination quantized average consensus algorithm with privacy-preserving guarantees. Furthermore, during the algorithm's operation each node processes and transmits quantized messages. We show that the algorithm converges in finite-time, and every node calculates the optimal allocation of test kits to infections. Additionally, we prove that, under specific conditions on the network topology, nodes are able to preserve the privacy of their initial state. Finally, we illustrate our proposed algorithm, present empirical simulation results regarding its convergence rate, and show how the utilized privacy-preservation mechanism affects the number of required time steps for convergence.

Index Terms—Distributed Networks, Optimal Resource Allocation, Privacy-Preservation, Distributed Optimization, Directed Graphs

I. INTRODUCTION

Over the last decades, various pandemics have posed challenges to healthcare and public health agencies in countries across the globe. One main challenge is to test individuals in the population for infection. Distributing test kits to various cities over a country is done according to the number of infections (e.g., a place with large number of infections requires more tests than one with less infections). However, some cities may also have concerns regarding fairness in the decision-making of the central public health agency as they need to ensure that tests are available in proportion to their infection numbers. Furthermore, gathering infection and test data into a central entity poses dangers for the entity to be a target of a hacking attack, which may expose sensitive health data regarding the number of infections of every city. Also, some cities may wish to preserve the privacy of their test kit possessions and infections from outside attackers or curious entities within other cities.

Consequently, there is a need for optimally allocating test kits in a distributed and privacy-preserving manner for controlling the spread of a pandemic. Optimal resource allocation is an optimization problem and aims to allocate resources such that specific performance objectives are satisfied. Optimal resource allocation has many applications such as optimally scheduling tasks for data centers [1], optimally coordinating the response of a set of distributed energy resources [2], optimally allocating vaccines/tests for pandemic stabilization [3], etc. Furthermore, there exists a variety of centralized algorithms for addressing optimal resource allocation problems (see for instance [4], [5]). However, a central entity could possibly suffer from processing issues due to network scale (i.e., the central entity may have to process vast amounts of data which imposes heavy computational requirements). Additionally, as mentioned previously, gathering all available data to a central entity may also impose privacy risks. For these reasons, in this paper we aim to address the optimal resource (i.e., test kit) 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 [6], [7]. However, a vast majority of the algorithms in the current literature assume that the messages exchanged among nodes consist of real values with infinite precision (see, for instance, [8]–[11]) and they exhibit asymptotic convergence within some error (see [12]). In practical applications, messages need to consist of quantized values (i.e., discrete values). Furthermore, the optimization procedure needs to calculate the exact solution in finite time in order for the calculated solution to be applicable in a realistic setting. Additionally, most optimal resource allocation algorithms typically do not provide privacy-preservation guarantees against curious entities (see [2], [13]). The latter is of particular importance in our case since, allocating test kits according to the number of infections may expose sensitive health data. Thus, there is a need for devising finite-time distributed optimal resource allocation algorithms with privacy-preserving guarantees.

In this paper, we discuss the distributed optimal resource allocation problem in the context of controlling the spread of a pandemic by optimally allocating test kits between cities (or individual entities) over a country. We consider that optimal distribution of test kits over a country is done according to the number of infections (i.e., a place with a large number of infections requires more test kits than one with less infections). A country can be modelled as a directed network of cities...
represented by nodes and communication links represented by directed edges. Furthermore, as in [13], we assume there are curious and private nodes. Curious nodes wish to infer the number of test kits and infections of other nodes and private nodes may modify the exchanged messages they transmit to prevent curious nodes from accessing their number of test kits and infections. We present a distributed privacy-preserving algorithm that optimally allocates test kits according to infections in a finite number of time steps. The operation of our algorithm is event-based and relies on processing and transmission of quantized messages. Exchange of quantized messages leads to more efficient communication and takes into consideration that the allocated resources (i.e., number of test kits) take quantized values. The privacy-preserving optimal test kit allocation algorithm, proposed in this paper, combines the approaches from [13] and [1] using the probabilistic protocol in [14] that exhibits faster convergence than most other quantized average consensus algorithms. Our main contributions are as follows.

- We present a probabilistic optimal allocation algorithm with efficient (quantized) communication and privacy-preservation guarantees. It is applied to distributed optimal test kit allocation over strongly connected networks; see Algorithm 1.
- We show that our privacy-preserving algorithm (i.e., Algorithm 1) deploys a distributed stopping mechanism to terminate its operation. Note that it is the first distributed stopping mechanism adjusted to the algorithm’s necessary privacy-preservation guarantees.
- We analyze theoretical guarantees for convergence of Algorithm 1 in particular the number of iteration steps required for all nodes to calculate the optimal allocation of test kits with positive lower bound probability; see Theorem 1.
- We provide sufficient topological conditions for privacy-preservation of Algorithm 1; see Theorem 2.

The optimal allocation algorithm in this paper uses properties of quantized average consensus algorithms [14]–[20], that allow nodes to exchange quantized messages. Furthermore, transmissions of quantized values are preformed asynchronously under event-triggered conditions, which increases the efficiency of communication. In particular, the optimal allocation algorithm proposed in this paper allows nodes to calculate a quantized ratio of test kits to infections, yielding a quantization error in contrast to using real finite-time average consensus algorithms [21]–[23]. However, assuming that the amount of available test kits is much greater than the number of infections, we obtain small relative quantization errors. The case of privacy preservation has been studied previously in [13], [24]–[28]. In particular, [13] 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 algorithm the injection of quantized offsets is performed according to a set of event-triggered conditions.

**Algorithm 1**

We present a probabilistic optimal allocation algorithm with efficient (quantized) communication and privacy-preservation guarantees. It is applied to distributed optimal test kit allocation over strongly connected networks. We show that our privacy-preserving algorithm (i.e., Algorithm 1) deploys a distributed stopping mechanism to terminate its operation. Note that it is the first distributed stopping mechanism adjusted to the algorithm’s necessary privacy-preservation guarantees. We analyze theoretical guarantees for convergence of Algorithm 1 in particular the number of iteration steps required for all nodes to calculate the optimal allocation of test kits with positive lower bound probability; see Theorem 1. We provide sufficient topological conditions for privacy-preservation of Algorithm 1; see Theorem 2.

**Fig. 1:** Example of nodes calculating the optimal resource allocation in a centralized fashion (i.e., via communication with a central entity).

**Notation and Graph Terminology**

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 real number \( a \in \mathbb{R} \), the floor is defined as \( \lfloor a \rfloor := \sup\{b \in \mathbb{Z} \mid b \leq a\} \) and the ceiling as \( \lceil a \rceil := \inf\{b \in \mathbb{Z} \mid b \geq a\} \). The communication network is represented by a strongly connected directed graph (digraph) \( G_d = (V, E) \) of \( n \) nodes. We denote the set of nodes by \( V \), the set of edges by \( E \) and the diameter by \( D \). For each node \( v_j \in V \) we denote the set of out-neighbours by \( N_{v_j}^+ \), in-neighbours by \( N_{v_j}^- \), out-degree by \( D_j^+ \) and in-degree by \( D_j^- \). We refer to [1] for standard definitions in graph theory.

**II. PROBLEM FORMULATION**

**A. Centralized Optimal Resource Allocation Problem**

The centralized optimal resource allocation problem is the following. Each node \( v_j \) is endowed with a scalar quadratic local cost function \( f_j : \mathbb{R} \mapsto \mathbb{R} \). Then, each node communicates with a central entity where it transmits stored data. The central entity (i) collects the data from every node in the network, (ii) calculates the optimal value which minimizes the sum of every node’s objective function, and (iii) transmits back to the nodes the calculated result. An example of the centralized optimal resource allocation is presented in Fig. 1 where the nodes communicate with a central entity (cloud) which provides them the value which minimizes the sum of their objective functions. In this paper, however, we are interested solving the optimal resource allocation problem in a distributed fashion.

**B. Distributed Optimal Resource Allocation Problem**

We now formally state the distributed optimal resource allocation problem as follows. For each node \( v_j \in V \), we define the scalar quadratic local cost function \( f_j : \mathbb{R} \mapsto \mathbb{R} \) as

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

where \( \alpha_j \in \mathbb{R}_+, \chi_j \in \mathbb{R}_+ \) is the demand at node \( v_j \), and \( z \) the global optimization parameter. Note that in the distributed version of the optimal allocation problem we consider that there is no central entity which receives and processes the data of every node in the network. We consider that every
node performs distributed calculations and message exchanges only with its direct neighbors. 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 interpretation of the global cost function is that it is the cost of all nodes in the network agreeing to obtain \( z \). Consequently, each node \( v_j \) aims to obtain a value \( z^* \) which minimizes the global cost function

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

Equation (2) has a closed form solution given by

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

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

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 allows 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 this as nodes that will try to infer the initial combined number of test kits and number of infections of non-curious nodes; \( V_p \) is the set of set of private nodes that want to preserve their privacy; \( V_n \) is the set of nodes that are neither curious nor private. We formulate the following second problem.

**Problem 1.** 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 Problem 1 we aim to find \( w_j^* \) such that

\[
    \frac{w_j^*}{\lambda_j} = q, \quad \forall v_j \in V
\]

where \( q = \frac{l_{\text{tot}} + u_{\text{tot}}}{\lambda_{\text{tot}}} \).

Note that \( q = (\sum_{v_i \in V} \lambda_{i})(l_i + u_i)/\lambda_i \) is the same as (5) with \( \alpha_j = \lambda_j \) and \( x_j = (l_i + u_i)/\lambda_i \) for all \( v_i \in V \). 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 [14] allows each node to calculate either the ceiling or the floor of \( q \) which yields the optimal allocation

\[
    w_j^* = \lfloor q \lambda_j - u_j \rfloor \text{ or } \lceil q \lambda_j - u_j \rfloor, \quad \forall v_j \in V.
\]

Equation (3) 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.

**C. Distributed Privacy-Preserving Optimal Resource Allocation Problem**

We now recall the following definition for privacy-preservation of a node.

**Definition 1.** A node \( v_j \) is said to preserve the privacy of its initial combined number of test kits \( l_j + u_j \) and its number of infections \( \lambda_j \) (or simply preserve its privacy) if these quantities cannot be inferred exactly by other nodes at any point during the operation of the algorithm.

Definition 1 serves as a basis for our second problem, wherein our focus is on addressing Problem 1 but under the constraint that some nodes are allowed to preserve their privacy while exchanging quantized information. Define the subsets \( V_p, V_c, V_n \subseteq V \) as follows [13]: \( V_c \) is the set of curious nodes that will try to infer the initial combined number of test kits and number of infections of non-curious nodes; \( V_p \) is the set of private nodes that want to preserve their privacy; \( V_n \) is the set of nodes that are neither curious nor private. We formulate the following second problem.

**Problem 2.** Solve Problem 1 under the conditions that every node \( v_j \in V_p \) (i) preserves the privacy of its initial state (i.e., number of test kits and number of infections), and (ii) it processes and transmits quantized information.

Problem 2 states that we want to prevent curious nodes from inferring the initial number of test kits and number of infections of private nodes. Nodes in \( V_p \) will follow additional instructions than nodes in \( V_n \) denoted as the privacy preservation mechanism the details are presented later in the paper. We assume curious nodes execute the same algorithm as nodes in \( V_n \), have knowledge of both algorithms (i.e., the algorithm which preserves privacy and the one which does not preserve privacy) and communicate/collaborate with each other. The case where curious nodes follow the same algorithm as nodes in \( V_p \) is an easier case of the problem we consider in this paper. Specifically, in this paper, curious nodes may collaborate arbitrarily to determine the initial state of the non-curious nodes. However, if the curious nodes follow the same algorithm as nodes in \( V_p \), then it means that they do not collaborate which facilitates privacy-preservation.

The following assumptions are required for the development of the main results in this paper.

**Assumption 1.** The communication network, which is modelled as a digraph, is strongly connected.
**Assumption 2.** The diameter $D$ (or an upper bound of $D$) is known to every node in the network.

Strong connectivity from Assumption 1 ensures that information transmitted by one node can reach every other node, and is important for guaranteeing convergence to the optimal solution. Furthermore, knowledge of an integer $C \geq D$, via Assumption 2 is required for terminating the operation of our algorithm once convergence has been achieved. Such an assumption is standard in the literature on consensus algorithms; see, for instance, [1], [29].

### III. Distributed Test Kit Allocation With Privacy-Preservation

In this section, we propose a quantized message exchange algorithm, namely Algorithm 1 that addresses Problem 2 (note that addressing Problem 2 means that we also address Problem 1). During its operation, the state of each private node is divided into two sub substates. Then, random offsets are injected into the transmitted messages. We also show that, under specific conditional policies, private nodes preserve the privacy of their initial state in addition to the network calculating the optimal allocation of test kits.

#### A. Distributed Algorithm for Optimal Test Kit Allocation With Privacy-Preservation

Algorithm 1 is executed by every node in the network. The set of nodes $\mathcal{V} \subseteq \mathbb{N}$ wherein execute an additional privacy preservation mechanism, preforming extra actions in order to preserve their privacy against curious nodes in $\mathcal{V}$. The intuition behind Algorithm 1 is the following. At every iteration step $k \in \mathbb{N}$ each node $v_j \in \mathcal{V}$ stores the mass variables $y_j[k], z_j[k]$. At initialization $k = 0$, they are set to $y_j[0] = I_j + u_j, z_j[0] = \lambda_j$. If $v_j \in \mathcal{V}_p$, $v_j$ randomly sets new mass variables $y_j^*_{[0]}, z_j^*_{[0]}$ and $y_j^*_{[1]}, z_j^*_{[1]}$, which are subject to $4z_j[0] = y_j^*_{[0]} + y_j^*_{[1]}$ and $4z_j[0] = z_j^*_{[0]} + z_j^*_{[1]}$, and offsets $\xi_{ij}, \zeta_{ij}$ for every $v_i \in \mathcal{N}_j^+$ such that $y_j^*_{[0]} = \sum_{v_i \in \mathcal{N}_j^+} \xi_{ij}, z_j^*_{[0]} = \sum_{v_i \in \mathcal{N}_j^+} \zeta_{ij}$. If $v_j \not\in \mathcal{V}_p$, $v_j$ instead defines $y_j^*_{[0]}, z_j^*_{[0]}$ such that $y_j^*_{[0]} = 4y_j[0]$ and $z_j^*_{[0]} = 4z_j[0]$. Define a token as a multisubst of two elements $\{y_{\tau}, z_{\tau} \in \mathbb{N}, \tau \in \mathbb{N}\}$, where $y_{\tau}$ denotes the token numerator and $z_{\tau}$ the token denominator. At every iteration step $k \in \mathbb{N}$, $v_j$ stores a list of tokens whose numerators and denominators form integer partitions of $y_j[k]$ and $z_j[k]$, respectively. If the event-triggered condition $z_j[k] > 0$ holds, $v_j$ randomly transmits stored tokens to its out-neighbours $v_i \in \mathcal{N}_j^+$, or keeps them. Node $v_j$ then receives tokens from its in-neighbours (or itself) and updates the mass variables in the next iteration step to be the sum of received token numerators and denominators [1], [14]. The main principle of the privacy preservation mechanism (i.e., lines 7 – 10, 16 – 17, 28 – 31 and 37 – 40 in Algorithm 1) is that if $v_j \in \mathcal{V}_p$ transmits a token to $v_i$ for the first time, $v_j$ adds random offsets to the numerator and denominator of that token. Furthermore, all nodes simultaneously follow a modified version of the synchronous max/min consensus algorithm [1], [29] that allow them to agree on the ceiling or the floor of the global ratio

![Algorithm 1](image)

**Output:** $w_j^*$ satisfies (6) for all $v_j \in \mathcal{V}$.

1. **Initialization:** Each node $v_j \in \mathcal{V}$ does the following:
   2. assigns a nonzero probability $b_{ij}$ to each node $v_i \in \mathcal{N}_j^+ \cup \{v_j\}$ as follows
   $$b_{ij} = \begin{cases} 
   \frac{1}{1+\xi_{ij}} & \text{if } v_j \in \mathcal{N}_j^+ \cup \{v_j\} \\
   0 & \text{otherwise}
   \end{cases}$$
   3. sets $y_j[0] = I_j + u_j, z_j[0] = \lambda_j$ and $\text{flag}_j = 0$.
   4. if $v_j \not\in \mathcal{V}_p$ then
   5. sets $y_j^*_{[0]} = 4y_j[0], z_j^*_{[0]} = 4z_j[0]$
   6. sets $\mathcal{N}_\text{sent}_j = \emptyset$
   7. else if $v_j \in \mathcal{V}_p$ then
   8. randomly chooses $y_j^*_{[0]}, y_j^*_{[1]}, z_j^*_{[1]}$ and $z_j^*_{[0]}$ such that $4y_j[0] = y_j^*_{[0]} + y_j^*_{[1]}$, $y_j^*_{[0]} \neq y_j^*_{[1]}$ and $4z_j[0] = z_j^*_{[0]} + z_j^*_{[1]}$, $z_j^*_{[0]} \neq z_j^*_{[1]}$
   9. for all $v_i \in \mathcal{N}_j^+$ defines $\xi_{ij}, \zeta_{ij}$ such that $y_j^*_{[0]} = \sum_{v_i \in \mathcal{N}_j^+} \xi_{ij}, z_j^*_{[0]} = \sum_{v_i \in \mathcal{N}_j^+} \zeta_{ij}$
   10. sets $\mathcal{N}_\text{sent}_j = \mathcal{N}_j^+$
   11. sets $\text{recv}_{ij}[0] = \bigcup_{R_j \in \mathcal{R}} \{\{\frac{y_j[0]}{z_j[0]}\}, 1\} \bigcup_{l_j \in \{0, 1\}} \{\{\frac{y_j[0]}{z_j[0]}\}, 1\}$ such that $y_j^*_{[0]} = \sum_{v_i \in \mathcal{N}_j^+} \xi_{ij}, z_j^*_{[0]} = \sum_{v_i \in \mathcal{N}_j^+} \zeta_{ij}$
   12. **Iteration:** For $k = 1, 2, \ldots$, each node $v_j \in \mathcal{V}$ does the following:
   13. while flag$_j = 0$ do
   14. if $k \equiv 1$ then
   15. sets $M_j = \max_{y_{\tau}, z_{\tau} \in \text{recv}_{ij}[k]} y$ and $m_j = \min_{y_{\tau}, z_{\tau} \in \text{recv}_{ij}[k]} y$
   16. if $v_j \not\in \mathcal{V}_p$ and $\mathcal{N}_\text{sent} \neq \emptyset$ then
   17. sets $M_j = M_j + 2$
   18. broadcasts $M_j, m_j$ to every $v_i \in \mathcal{N}_j^+$
   19. receives $M_j, m_j$ from every $v_i \in \mathcal{N}_j^-$
   20. sets $M_j = \max_{v_i \in \mathcal{N}_j^+} M_j$ and $m_j = \min_{v_i \in \mathcal{N}_j^-} m_j$
   21. sets $\text{recv}_{ij}[k] = \emptyset$ and $c_{ij}^k[0] = 0, c_{ij}^k[1] = 0$ for every $v_i \in \mathcal{N}_j^+ \cup \{v_j\}$
   22. if $z_j^*_{[0]} > 1$ then
   23. sets $y_j^*_{[k]} = y_j^*_{[k]}, z_j^*_{[k]} = z_j^*_{[k]}, q_{ij}^k[0] = \frac{y_j^*_{[k]}}{z_j^*_{[k]}}$
   24. removes $\{y_j^*_{[k]}, z_j^*_{[k]}\} \in \text{recv}_{ij}[k]$ s.t.
   25. sets $\text{recv}_{ij}[k] = \text{recv}_{ij}[k] \cup \{y_j^*_{[k]}, z_j^*_{[k]}\}$, $c_{ij}^k[0] = c_{ij}^k[0] + y_j^*_{[k]}$, $c_{ij}^k[1] = c_{ij}^k[1] + z_j^*_{[k]}$
   26. while $\text{recv}_{ij}[k] \neq \emptyset$ do
   27. chooses $v_i \in \mathcal{N}_j^+ \cup \{v_j\}$ randomly according to $b_{ij}$ and removes $\{y_j^*_{[k]}, z_j^*_{[k]}\} \in \text{recv}_{ij}[k]$
   28. if $v_j \not\in \mathcal{V}_p$ and $v_i \in \mathcal{N}_\text{sent}$ then
   29. (i) $\{y_j^*_{[k]}, z_j^*_{[k]}\} = \{y_j^*_{[k]} + \xi_{ij}, q_{ij}^k[0] + \zeta_{ij}, \}$, (ii) $y_j^*_{[k]} = y_j^*_{[k]} - \xi_{ij}, z_j^*_{[k]} + 1 = z_j^*_{[k]} - \zeta_{ij}$, (iii) $\mathcal{N}_\text{sent}_j = \mathcal{N}_\text{sent}_j \setminus \{v_i\}$
of test kits in infections in the network after finite time (i.e., calculate the optimal allocation only if the privacy preservation mechanism has completed).

We state the following additional assumption which guarantees that tokens present in the network have, at all iteration steps, numerators greater than or equal to their denominators (e.g., the global number of test kits is greater than the global number of infections in the network). This assumption is necessary for the convergence of Algorithm 1.

**Assumption 3.** The initialization of Algorithm 1 for each node $v_j \in V$ is such that

1. $y_j[0] \geq z_j[0]$,
2. if $v_j \in V_p$, then $y_j[0] \geq z_j[0] \geq 2$, and
3. $\xi_{ij} \geq \xi_{ij} \geq 0$ for all $v_i \in N^+_j$.

**Remark 1.** Note here that the Algorithm 1 addresses both Problem 1 and Problem 2 presented in Section 1. However, the operation of Algorithm 1 can be adjusted to solve only Problem 1 by removing the privacy preservation mechanism (i.e., lines 7–10, 16–17, 28–31 and 37–40). In this way, Algorithm 1 optimally allocates test kits without privacy preservation guarantees (an example is shown in [12] which aims to balance the CPU utilization across data center servers by deciding how to allocate CPU resources to workloads in a distributed fashion).

### B. Guarantees for Convergence and Privacy Preservation

Now we show that during the operation of Algorithm 1 each node is able to calculate the optimal allocation of test kits in a distributed and privacy-preserving fashion and then terminate its operation. We first state Theorem 1 which analyzes the finite-time convergence of Algorithm 1.

**Theorem 1.** Consider a digraph $G_d = (V, E)$ under Assumptions 1 to 3. Let $u_j$ be the number of stored test kits, $l_j$ the number of received test kits, and $\lambda_j$ the number of infections at each node $v_j \in V$. Suppose that each node $v_j$ follows the initialization and iteration steps as described in Algorithm 1. For any $\epsilon_1, \epsilon_2 \in (0, 1)$ there exist iteration steps $k_0^l, k_0^r \in \mathbb{N}$ such that each node $v_j \in V$ calculates the optimal allocation of test kits $w_j^*$ (as shown in (6)) after $k_0^l$ iteration steps with probability at least $(1 - \epsilon_2)(\epsilon_1^{q_{init} + 1}) \prod_{v_i \in V_p} (1 - \epsilon_1)^{D^+} (1 + D^+)^{-D^+}$, where $q$ satisfies (5) and

$$y_j^{\text{init}} = \sum_{v_i \in V} \left( \left\{ \frac{y_j[k_0]}{z_j[k_0]} \right\} - [q] \right) + \sum_{v_i \in V_p} \left( [q] - \frac{y_j[k_0]}{z_j[k_0]} \right).$$

Thereafter, the nodes terminate the iteration.

**Proof:** See Appendix. $\blacksquare$

The following theorem provides topological conditions for privacy preservation. It states that a private node preserves its privacy provided it has a private in-neighbour or out-neighbour. Together with Theorem 1 this solves Problem 2.

**Theorem 2.** Suppose that the conditions in Theorem 1 are satisfied. Additionally, assume there exists a subset of nodes $V_p \subseteq V$, $|V_p| \geq 2$ that follow the privacy preservation mechanism of Algorithm 1. Then $v_j \in V_p$ preserves the privacy of its initial state if it has at least one in-neighbour or out-neighbour $v_i \in V_p$.

**Proof:** See Appendix. $\blacksquare$

### IV. Simulation Results

We now present simulation results of Algorithm 1 for the setting where a disease is spreading in a country and test kits need to be optimally allocated across different cities depending on the number of infections of each city. The operation of Algorithm 1 is demonstrated in Fig. 2 and Fig. 3, where its rate of convergence and the mean number of iterations required for convergence with and without the privacy preservation mechanism, are shown, respectively.

Figure 2 shows the convergence of Algorithm 1. In Fig. 2a each line represents the state variable $q_j[k]$ corresponding to the calculated value of the global ratio $q$ in (5) at each node for every iteration step. At termination, all state variables have converged to either the ceiling or the floor of $q$. Fig. 2b shows the corresponding calculation of the optimal allocation where each line is represented by $w_j^* + u_j = q_j[k] \lambda_j$ at every iteration step. Random choices of the initial number of test kits and infections were made such that $l_j + u_j \in [200, 400]$, and $\lambda_j = 1$ or 2 for all $v_j \in V$ (which explains the convergence in
Vaccine Allocation for Pandemic Stabilization

Calculation of the global ratio of tests to cases

Fig. 2: Convergence towards the global ratio of test kits to infections marked as a red star (*) (in (a)) and optimal allocation marked as blue stars (*) in (b) in a random strongly connected network of 100 nodes under Algorithm 1. Each line represents the state variable $q_j^s[k]$ in the top plot and $w_j^s[k] + u_j = q_j^s[k] \lambda_j$ in the bottom plot for each $v_j \in \mathcal{V}$.

Calculation of the optimal allocation of tests

Mean iteration length as a function of $\lambda$

Fig. 3: Comparison between the number of iterations required for Algorithm 1 with and without the privacy preservation mechanism to converge for a network of 10 nodes (in (a)) and 100 nodes (in (b)) when varying the number of cases $\lambda_j$ which is set equal at all nodes. The number of test kits $l_j + u_j$ is randomly set in the interval $[500, 1500]$ at each node. In the case of 10 nodes and for $\lambda_j$ greater than 15 infections in the 100 nodes case, both algorithms (i.e., with and without privacy preservation) have approximately the same iteration length. However, in the 100 nodes case, Algorithm 1 takes notably longer to converge than Algorithm 1 without the privacy preservation mechanism for small $\lambda_j$ considering around 1–10 infections. Note here that in practice, we would most likely wish to find the optimal allocation of test kits when cities in a country experience more than 15 cases. This means that as long as the considered network is not much greater than 100 cities, every city may preserve its privacy without any noticeable loss in computation time.

V. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we presented a distributed privacy-preserving algorithm that optimally allocates test kits according 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 and then terminate their operation. We also provided sufficient topological conditions for privacy-preservation where each node is able to preserve the privacy of its initial number of test kits and infections. In our simulation results we demonstrated our proposed algorithm to the allocation of test kits according to infections for controlling the spread of a pandemic. There we showed that the rate of convergence for the algorithm with and without privacy-preservation is approximately the same for medium sized networks.

REFERENCES

[1] A. I. Rikos, A. Grammenos, E. Kalyvianaki, C. N. Hadjicostis, T. Charalambous, and K. H. Johansson, “Optimal cpu scheduling in data centers via a finite-time distributed quantized coordination mechanism,” arXiv preprint arXiv:2104.03126, 2021.

[2] M. Zholbaryssov, C. N. Hadjicostis, and A. D. Dominguez-Garcia, “Privacy-preserving distributed coordination of distributed energy resources,” in 59th IEEE Conference on Decision and Control (CDC), 2020, pp. 4689–4696.

[3] Y. Ma, Y.-Y. Liu, and A. Olshevsky, “Optimal vaccine allocation for pandemic stabilization,” 2021. [Online]. Available: https://www.researchgate.net/publication/354542774_Optimal_Vaccine_Allocation_for_Pandemic_Stabilization.

[4] L. Fang, “A generalized dea model for centralized resource allocation,” European Journal of Operational Research, vol. 228, no. 2, pp. 405–412, 2013.

[5] F. H. Lotfi, N. Nematiollahi, M. H. Behzadi, M. Mirbolouki, and Z. Moghaddas, “Centralized resource allocation with stochastic data,” Journal of Computational and Applied Mathematics, vol. 236, no. 7, pp. 1783–1788, 2012.

[6] A. Nedić, A. Olshevsky, and M. G. Rabbat, “Network topology and communication-computation tradeoffs in decentralized optimization,” Proceedings of the IEEE, vol. 106, no. 5, pp. 953–976, 2018.

[7] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, “A survey of distributed optimization,” Annual Reviews in Control, vol. 47, pp. 278–305, 2019.
Lemma 1. Suppose that the conditions in Theorem 2 are satisfied. For any $\epsilon \in (0,1)$ there exists an iteration step $k_0' \in \mathbb{N}$ such that the following hold with probability at least $\prod_{v_j \in  V_p} (1 - \epsilon)^{-D^+_j} (1 + D^+_j)^{-D^-_j}$: (i) The sums of offsets $\sum_{v_j \in  V_p} y_j^0[0], \sum_{v_j \in  V_p} z_j^0[0]$ have each been added to the corresponding sums of mass variables $\sum_{v_j \in  V_p} y_j^0[k_0], \sum_{v_j \in  V_p} z_j^0[k_0]$ and (ii) $y_j^0[k] \geq z_j^0[k]$ for all nodes $v_j \in  V$ and iteration steps $k \in \mathbb{N}$ (in particular for $k = k_0'$).

Proof of Lemma 7
First note that the iteration of Algorithm 1 will not stop for any node $v_j \in  V$ before all private nodes have completed the privacy preservation mechanism (i.e., have injected the calculated offsets for every out-neighbour - seen in lines 26–33 of Algorithm 1). In particular, the lines 16 – 17 prevent the convergence of the max/min consensus algorithm (seen in lines 14 – 20 and 41 – 42) and consequently the termination of Algorithm 1 as long as there remain out-neighbours of private nodes that have not yet received a token with an added offset.

To prove statement (i), it suffices to show that we can always construct an iteration step $k_0' \in \mathbb{N}$ such that every out-neighbour of all nodes that want to preserve their privacy have received a token and hence an offset with nonzero probability. By items (i) and (ii) in Assumption 3 the iteration is initialized by each node transmitting tokens to its out-neighbours or to itself. If all nodes initially have more tokens than out-neighbours, i.e., $z_j^0[0] \geq 1 + D^+_j$ for all $v_j \in  V_p$, then, after the first iteration, all out-neighbours of $v_j$ receive a token with an added offset with probability at least $(1 + D^+_j)^{-D^-_j}$, and this holds for all $v_j \in  V_p$ with probability at least $\prod_{v_j \in  V_p} (1 + D^+_j)^{-D^-_j}$. Otherwise, there exists a nonempty subset of nodes $S \subseteq  V_p$ with $z_j^0[0] \leq D^-_j$ for all $v_j \in  S$. Item (iii) in Assumption 3 now ensures that every node that receives tokens will transmit them in the next iteration. From the proof of Theorem 3 and Assumption 1 it follows that for any $\epsilon \in (0,1)$ the probability that a token starting from a node $v_k$ at node $v_k$ after $\tau(n-1)$ iteration steps for any different two nodes $v_j, v_k \in  V$ is $P_{T^{out}} \geq 1 - \epsilon$ where $\tau = \left[ \frac{\log \epsilon}{\log(1-(1+D^+_j)(n-1))} \right]$, $D^+_j \leq \frac{\max_{v_j \in  V_p} D^+_j}{v_j \in  V_p}$.

Furthermore, items (i) and (ii) in Assumption 3 imply there are more tokens in the entire network preforming a random walk than nodes, and hence also more tokens preforming a random walk than $D^-_j \leq n-1$ at every iteration step $k \in \mathbb{N}$. Additionally, every node transmits a token to itself (i.e., keeps a token) at every iteration step, which implies that $z_j^0[k] \geq 1$ for all $v_j \in  V, k \in \mathbb{N}$. Consequently, for any $\epsilon \in (0,1)$, there

1Applicable here since the tokens of Algorithm 1 also preform a random walk.

APPENDIX

We first present Lemma 1 which is useful for proving Theorem 1 and Theorem 2. Lemma 1 shows that Assumptions 1 provide a sufficient condition for ensuring that all $v_j \in  V_p$ preserve their privacy during the operation of Algorithm 1.
are a minimum of $D_j^+ + 1$ tokens at a node $v_j \in \mathcal{S}$ after $\tau(n - 1)$ iteration steps with probability at least $(1 - \epsilon)^{D_j^+}$, which means all out-neighbours of $v_j$ receive a token with an added offset after $\tau(n - 1) + 1$ iteration steps with probability at least $(1 - \epsilon)^{D_j^+} (1 + D_j^+) - D_j^+$. Hence, for any $\epsilon \in (0, 1)$, all out-neighbours of every $v_j \in \mathcal{V}_p$ receive an offset after $k_0'| = |S|[\tau(n - 1) + 1]$ iteration steps with probability at least $\prod_{v_j \in \mathcal{V}_p} (1 - \epsilon)^{D_j^+} (1 + D_j^+) - D_j^+$, which proves statement (i) in this lemma.

To prove statement (ii), we assume that the total sums of offsets have been added to the network at iteration step $k_0$. We note that
\[
\sum_{v_j \in \mathcal{V}} y_j^\alpha[k_0] = \frac{\sum_{v_j \in \mathcal{V}} 4y_j[0]}{\sum_{v_j \in \mathcal{V}} 4z_j[0]} = \frac{\sum_{v_j \in \mathcal{V}} y_j[0]}{\sum_{v_j \in \mathcal{V}} z_j[0]} = \lambda_{tot} + u_{tot} = q.
\]
holds at iteration step $k_0'$, where the last equality is due to (5), which proves statement (ii) in this lemma.

To prove statement (iii), we first note that $y_j^\alpha[0] \geq z_j^\alpha[0]$ for all $v_j \in \mathcal{V}$ at initialization due to items (i) and (ii) in Assumption 2. Hence, tokens present in the network at initialization will have numerators greater or equal than their denominators, which also holds for all iteration steps $k \in \mathbb{N}$ due to the injection of offsets satisfying item (iii) in Assumption 3. Since the mass variables $y_j^\alpha[k], z_j^\alpha[k]$ are sums of token numerators and denominators, we have that $y_j^\alpha[k] \geq z_j^\alpha[k]$ for all $k \in \mathbb{N}$ which proves statement (iii) in this lemma. ■

**Proof of Theorem 2**

We also need the following result to prove Theorem 2.

**Proposition 1 (Proposition 2).** Consider a digraph $G_d = (\mathcal{V}, \mathcal{E})$ under Assumptions 1 and item (2) in Assumption 2. Let $\ell_j$ be the number of stored test kits, $\ell_j$ the number of received test kits and $\lambda_j$ the number of injections at each node $v_j \in \mathcal{V}$. During the operation of Algorithm 2 without the privacy preservation mechanism (i.e., without lines 7–10, 16–17, 28–31 and 37–40), for any $\ell_2 \in (0, 1)$ there exists an iteration step $k_0 \in \mathbb{N}$ such that each node $v_j$ calculates the optimal allocation of test kits $w_j^\beta$ (as shown in (6)) with probability at least $(1 - \epsilon_2)(\gamma^\ell \beta + \beta)$ where
\[
y_j^\text{init} = \sum_{v_j \in \mathcal{V}} \left(\left\lfloor \frac{y_j[0]}{z_j[0]} \right\rfloor - \left\lfloor q \right\rfloor \right)
+ \sum_{v_j \in \mathcal{V}} \left(\left\lfloor q \right\rfloor - \left\lfloor \frac{y_j[0]}{z_j[0]} \right\rfloor \right),
\]
where $q$ is defined in (5). Thereafter, the nodes terminate the iteration.

We now present the proof of Theorem 2.

**Proof of Theorem 2** Note that in Proposition 1 we considered Algorithm 1 without the privacy preservation mechanism (i.e., without lines 7–10, 16–17, 28–31 and 37–40). According to Lemma 1 there exists an iteration step $k_0'$ in the operation of Algorithm 1 such that all private nodes have completed the privacy preservation mechanism, which means that during iteration steps $k > k_0'$ the operation of Algorithm 1 does not execute the privacy preservation steps. Additionally, note that in Proposition 1 we considered item (2) in Assumption 3 which states that, in Algorithm 1 without the privacy preservation mechanism, the mass variables satisfy $y_j[0] \geq z_j[0]$ at initialization step $k = 0$ for all nodes $v_j \in \mathcal{V}$. According to Lemma 1 there exists an iteration step $k_0$ in Algorithm 1 such that $y_j^\beta[k_0'] \geq z_j^\beta[k_0]$ hold for all $v_j \in \mathcal{V}$, which corresponds to item (2) in Assumption 3 instead at iteration step $k = k_0$. Therefore, we may use the result of Proposition 1 to Algorithm 1 for iteration steps $k > k_0'$, replacing $y_j^\alpha$ by the one stated in Theorem 1.

The above argument in combination with the probabilistic convergence from Lemma 1 imply that under the initialization and iteration of Algorithm 1 for any $\epsilon_1, \epsilon_2 \in (0, 1)$, each node calculates the optimal allocation of test kits after $k_0'' = k_0 + k_0'$ iteration steps with probability at least $(1 - \epsilon_2)\gamma^\ell \beta + \beta \prod_{v_j \in \mathcal{V}_p} (1 - \epsilon_1)^{D_j^+} (1 + D_j^+) - D_j^+$ where $k_0$ is the iteration step in Proposition 1 dependent on $\epsilon_2$, $k_0'' = \max\{|S|[\tau(n - 1) + 1], 1\}, \gamma_0 \geq \left\lceil \frac{\log(1 - \epsilon_1)\gamma^\ell \beta}{\log(1 - (\epsilon_1 + \epsilon_2))}\right\rceil$, $S = \{v_j \in \mathcal{V}_p | z_j^\beta[0] \leq D_j^+\}$ and then terminate its operation.

**Proof of Theorem 2**

Assume $v_j \in \mathcal{V}_p$ has at least one in-neighbour $v_i \in \mathcal{V}_p$. It suffices to show that $v_j$ preserves its privacy if all other in-neighbours and out-neighbours of $v_j$ and $v_i$ are curious. We have that $v_j$ will receive a token with the offsets $\xi_j, \zeta_j$ from $v_i$ and add the offsets $\xi_j, \zeta_j$ to tokens it transmits for all $v_j \in \mathcal{N}_j$. We consider the two possible scenarios.

A. If not every $v_j \in \mathcal{N}_j$ has received a token in the first iteration step $k = 1$, then the curious nodes know $v_j$ may receive a token with an added offset from $v_i$ at any iteration step before $v_j$ completes the privacy preservation mechanism. Hence, they cannot determine whether or not a detected offset in a token received from $v_j$ was injected by $v_j$ or $v_i$. Therefore, they cannot exactly infer the value of $y_j^\beta[0], z_j^\beta[0]$ nor $y_j^\beta[0], z_j^\beta[0]$. 

B. If every $v_j \in \mathcal{N}_j$ receives a token at $k = 1$, all curious out-neighbours know they have received all the offsets of $v_j$. Similarly to the previous case, $v_j$ will transmit a token with a random offset to $v_j$. Additionally, $v_j$ may transmit tokens to itself at any iteration step. This means that the curious nodes cannot determine whether or not the offset received by $v_j$ from $v_i$ was part of $v_j$'s initial mass variables $y_j^\beta[0], z_j^\beta[0]$. Therefore, they cannot exactly infer $y_j^\beta[0], z_j^\beta[0]$ nor $y_j^\beta[0], z_j^\beta[0]$. 

In both cases, since $4y_j[0] = y_j^\beta[0] + y_j^\delta[0], 4z_j[0] = y_j^\alpha[0] + y_j^\delta[0]$ for $k = j$ or $i$, we have that both $v_j$ and $v_i$ preserve the privacy of their initial states $y_j[0], z_j[0]$ for $k = i$ or $j$.

By symmetry, the same result holds if $v_j \in \mathcal{V}_p$ has at least one out-neighbour $v_i \in \mathcal{V}_p$ and all other in-neighbours and out-neighbours of $v_j$ and $v_i$ are curious. ■