A Signed-Message Relay Version of Iterative Approximate Byzantine Consensus for Directed Graphs

Matthew Ding
PRIMES CS/Bio Spring Conference
23 May 2021
The Byzantine Generals Problem

- Paper published by Lamport, Shostak, and Pease in 1982
- Coined the term “byzantine fault”: a machine that can arbitrarily deviate from an agreed upon protocol in opposition to other users
- Very strict assumption, but sometimes necessary in real life
Problem Formulation

- Group of generals camped outside of a city
- Each general may communicate with each other individually, but no centralized communication exists
- The goal is for each general to decide on the same course of action: either “attack” or “retreat”
- There exist secret “byzantine generals”, who may act arbitrarily and whose goal is to prevent “honest generals” from achieving their goals
A More Mathematical Representation

- Byzantine consensus problems are usually represented as graphs.
- Nodes represent generals, edges represent communication links.
Iterative Approximate Byzantine Consensus (IABC)

• Each honest node holds a real number value as their current state

• Honest nodes achieve approximate consensus on their states with one another rather than exact consensus

• Aim to satisfy two conditions:
  1. Convergence: Every honest node’s state approaches the same value as the number of iterations approaches infinity
  2. Validity: Each honest node’s state in each iteration remains within the convex hull of states of the previous iteration
Existing IABC Algorithm

• Developed by Vaidya in 2012
• During each iteration, each honest node transmits their current state to all neighbors
• Each honest node performs a trimmed-mean step to determine its new state for the next iteration
• Proven that each honest node achieves consensus on the same value over time
Trimmed-Mean Step

- Given a list of at least $3f+1$ values:
  1. Sort the list
  2. Eliminate the greatest and least $f$ values
  3. Output the arithmetic mean of the remaining values

- This is a robust aggregation step for up to $f$ byzantine nodes
Our Contributions

• Signatures
  • Incredibly important in Byzantine consensus, but new to IABC
  • Reliable proof of who created a message

• Relays
  • Using signatures, we can now reliably relay messages across a graph
  • Even if a message has been relayed across multiple nodes, we can reliably detect the node of origin
Our Contributions (continued)

• With relays and signatures, nodes don’t need to be adjacent to communicate with each other

• All honest nodes in a graph may send and receive messages to every other honest node

• Our algorithm creates a “pseudo-complete” graph in order to increase the efficiency of communication across the graph
3.4 Relay-IABC Algorithm

Algorithm 1: Relay-IABC

Result: Each state \( v_i(t) \) remains within the convex hull of the initial states at each iteration, and each state converges to the same value as iteration \( t \to \infty \).

Initialization:
\[ v_i(0) = \text{Initial State of node } i \text{ (with signature } i) \]

for \( t = 0 \) to \( T \) do
    Broadcast \( v_i \) to all machines \( j \in N_G^D \)
    Receive \( v_j \) from all machines \( j \in N_G^D \)

Remark. When receiving \( v_j \), ignore all parameters received that are not properly signed. If no proper message is received from a certain node, set their incoming value to be an arbitrary predefined real value (e.g. 0).

\[ G_t = N_G^D \cup \{i\} \]

for \( j = 0 \) to \( m - 1 \) do
    if \( j \neq i \) then
        \( v_i(j) \leftarrow v_j(j) \)
    end
end

if \( t \mod D = 0 \) then
    Trimmed-mean update step:
    In a new vector, sort the values of \( v_i \) in increasing order:
    \[ v_i^* \leftarrow \text{sort}(v_i) \]
    Ignore the least and greatest \( b \) values, and set the value of \( v_i(i) \) to be the average of all remaining values in \( v_i^* \), as defined below:
    \[ v_i(i) \leftarrow \frac{1}{m - 2b} \sum_{k=b}^{m-b} v_i^*(k) \]
    Add signature \( i \) to \( v_i(i) \)
end

• Every honest node stores most recent state values of every other node
• In each iteration, every honest node relays state values of every node to all neighbors
• Each state value is tagged with signature
• Trimmed-mean is used with the state values of all nodes instead of just neighbors
• Perform trimmed-mean step only every \( d \) iterations, where \( d \) is the diameter of the honest subgraph
Theoretical Convergence Rate

• Original IABC algorithm
  • Non-zero column in $M^{rh}$

• Relay-IABC algorithm
  • Non-zero column in $M^3$
  • d times more iterations per $M$, but net convergence is faster

• $(1 - \varepsilon^d)^T \gg d(1 - \varepsilon)^T$
Simulation Results

- Compares IABC and Relay-IABC convergence rates
- Relay-IABC achieves faster convergence

Simulation Graph: Network of 30 honest nodes, 14 byzantine nodes
Net Benefits and Tradeoffs

• Relay-IABC achieves convergence on sparse graphs – better represent “real-world” networks

• Even on dense networks, we show empirically that Relay-IABC achieves faster convergence in large variety of cases

• Tradeoff: higher communication cost
Future Work

• Relationship between update frequency and convergence rate

• Tolerating a higher proportion of Byzantine nodes (signatures)
Acknowledgements

• MIT PRIMES
• Hanshen Xiao
• My parents

Thank you!