The Power of Distributed Verifiers in Interactive Proofs

Abstract:

We explore the power of interactive proofs with a distributed verifier. In this setting, the verifier consists of n nodes and a graph G that defines their communication pattern. The prover is a single entity that communicates with all nodes by short messages. The goal is to verify that the graph G belongs to some language in a small number of rounds, and with small communication bound, i.e., the proof size.

This interactive model was introduced by Kol, Oshman and Saxena (PODC 2018) as a generalization of non-interactive distributed proofs. They demonstrated the power of interaction in this setting by constructing protocols for problems as Graph Symmetry and Graph Non-Isomorphism -- both of which require proofs of (n^2)-bits without interaction.

In this work, we provide a new general framework for distributed interactive proofs that allows one to translate standard interactive protocols (i.e., with a centralized verifier) to ones where the verifier is distributed with a proof size that depends on the computational complexity of the verification algorithm run by the centralized verifier. We show the following:

* Every (centralized) computation performed in time O(n) on a RAM can be translated into three-round distributed interactive protocol with O(log n) proof size. This implies that many graph problems for sparse graphs have succinct proofs (e.g., testing planarity).

* Every (centralized) computation implemented by either a small space or by uniform NC circuit can be translated into a distributed protocol with O(1) rounds and O(log n) bits proof size for the low space case and polylog(n) many rounds and proof size for NC.

* We show that for Graph Non-Isomorphism, one of the striking demonstrations of the power of interaction, there is a 4-round protocol with O(log n) proof size, improving upon the O(n*log n) proof
size of Kol et al.

* For many problems, we show how to reduce proof size below the seemingly natural barrier of \log n. By employing our RAM compiler, we get a 5-round protocol with proof size \( O(\log \log n) \) for a family of problems including Fixed Automorphism, Clique and Leader Election (for the latter two problems we actually get \( O(1) \) proof size).

* Finally, we discuss how to make these proofs non-interactive \( \text{em arguments} \) via random oracles.

Our compilers capture many natural problems and demonstrate the difficulty in showing lower bounds in these regimes.
Joint work with Moni Naor and Merav Parter.