Congestion-Free Rerouting of Flows on DAGs

Saeed Akhoondian Amiri
Technical University Berlin

Dec 7, 2016, 13:00 p.m

We initiate the theoretical study of a fundamental practical problem: how to schedule the congestion-free rerouting of k flows? The input to our problem are k path pairs: for each of the k unsplittable flows (of a certain demand), there is an old and a new path along which the flow should be routed. As different flows can interfere on the physical links, the updates of the different flows at the different nodes must be scheduled such that transient congestion is avoided. This optimization problem finds immediate applications, e.g., in traffic engineering in computer networks. We show that the problem is generally NP-hard already for $k = 2$ flows. Interestingly, we find that for general k, deciding whether an unsplittable multi-commodity flow rerouting schedule exists, is even NP-hard on DAGs. Both NP-hardness proofs are non-trivial. We then present two polynomial-time algorithms to solve the route update problem for a constant number of flows on DAGs. Both algorithms employ a decomposition of the flow graph into smaller parts which we call blocks. Based on the given block decomposition, we define a dependency graph whose properties can be leveraged to compute an optimal solution for $k = 2$ flows. For arbitrary but fixed k, we introduce a weaker dependency graph and present our main contribution: an elegant linear-time algorithm which solves the problem in time $2^{O(k \log k)} O(|G|)$.