A MEMETIC ALGORITHM FOR OPTIMIZING ROUTING IN NETWORKS USING EXPONENTIAL FLOW SPLITTING

Mauricio G. C. Resende¹, Roger Reis², Marcus Ritt³, and Luciana S. Buriol⁴
¹AT&T Labs Research, Florham Park, NJ 07932-0971, mgcr@research.att.com; ²,³,⁴Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, 91501-970, Brazil, {roger.reis,mrpritt,buriol}@inf.ufrgs.br

Abstract: Routing protocols like OSPF (Open Shortest Path First) work on a set of link weights set up to control network data flow. Finding link weights that minimize network congestion for a given network topology and demand matrix is known as the Weight Setting Problem (WSP) and was proved to be NP-Hard. A new protocol called Distributed Exponentially-weighted Flow Splitting, or DEFT, has been recently proposed by Xu, Chiang, and Rexford (2007). DEFT with real-valued weights was shown to outperform OSPF with integer weights and often obtains optimal or near-optimal congestion. In this paper, we propose a memetic algorithm for the WSP for DEFT. Our algorithm can be used as part of a network design tool. It is restricted to using integer weights. We compare results with the two-stage iterative method of Xu, Chiang, and Rexford and observe that the two-stage method finds real-valued weights that lead to less congestion than the integer weights of the memetic algorithm. We propose an extension of our algorithm to deal with real valued weights.

DETECTING CRITICAL NODES IN SPARSE GRAPHS

Ashwin Arulselvan¹, Clayton W. Commander², and Panos M. Pardalos³
¹Dept. of Industrial & Systems Eng, Univ. of Florida, Gainesville, FL 32611 {ashwin,pardalos}@ufl.edu; ²Air Force Research Laboratory, Eglin AFB, FL 32542, clayton.commander@eglin.af.mil

Abstract: Identifying critical nodes in a graph is important to understand the structural characteristics and the connectivity properties of the network. In this paper, we focus on detecting critical nodes, or nodes whose deletion results in maximum disconnectivity of the graph. This problem, known as the CRITICAL NODE DETECTION PROBLEM has applications in several fields including biomedicine, telecommunications, and military strategic planning. In this talk we discuss two formulations of the problem and the computational complexity of each. Further we present the numerical results of heuristic algorithms which have been implemented for each. The solutions are compared with those found by a commercial software package.