Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 52 (2016) 367–374 www.elsevier.com/locate/endm

An Approximation Algorithm for Multiroute Flow Decomposition

Vorapong Suppakitpaisarn [1,2]

Global Research Center for Big Data Mathematics,
National Institute of Informatics;
and ERATO Kawarabayashi Large Graph Project,
Japan Science and Technology Agency, Tokyo, Japan

Abstract

In this paper, we give an algorithm to optimize the number of elementary k-flows obtained by decomposing a given k-route flow. It is known that a k-route flow can be decomposed into a linear combination of elementary k-flows, and there are many algorithms proposed for that decomposition. However, the number of elementary k-flows from those algorithms can be as large as $O(|E|)$ when $|E|$ is the number of edges in our network. As we have to reroute our flow for each elementary k-flow, it is more desirable to have a smaller number of the elementary flows in the combination. Let v be a maximum k-route flow value of our network, and h be a real number such that $0 < h < 1$. Denote τ_h be the maximum value such that the flow value of the network is $h \cdot v$, when edges with flow value less than τ_h are removed. We propose an algorithm to decompose a $(v - k \cdot \tau_h \cdot OPT)/v$-fraction of a k-route flow to at most $\frac{1}{1-h} \cdot OPT$ elementary k-flows, when OPT is an optimal number of elementary k-flows required.

Keywords: graph optimization algorithm, network flow, multiroute flow, flow decomposition, robust communication

http://dx.doi.org/10.1016/j.endm.2016.03.048
1571-0653/

1 Introduction

In this paper, we aim to optimize an efficiency of *k-route flow*, which is a tool for robust network communication. The tool is proposed by Kishimoto and Takeuchi in [9], before it is generalized by Kishimoto in [7,8]. We can consider a flow as a linear combination of *elementary k-flows*. Those elementary k-flows are network flows in which the flow value at all edges are 0 or 1, and the set of edges with flow value 1 is a k-edge disjoint paths from a source node to a sink node. In [9], Kishimoto also propose an algorithm for finding a maximum k-route flow, a k-route flow which has the largest flow value, for a given network. The algorithm is later improved in [1] by Aggarwal and Orlin.

The k-route flow has been used for a robust communication in many application domains. Those applications include maximum barrier coverage in [10,3,11] and k-MLA-decomposition in [2]. Beside the maximum k-route flow, we also need a linear combination of elementary k-flows in those applications. Denote the number of links in the given network by $|E|$. Kishimoto has proposed an $O(|E|^4)$ algorithm to find that linear combination in [6]. The number of elementary k-flows in the combination obtained from the algorithm is guaranteed to be in $O(|E|^2)$. Later, Aggarwal and Orlin improve the computation time to $O(|E|^3)$ [3] and improve the number of elementary k-flows to $O(|E|)$.

Although $O(|E|)$ is small enough for many applications, many implementations require a smaller number of elementary k-flows in the linear combination. For the maximum barrier coverage problem, the number of times we have to reroute our communication flow between a source node and a sink node is equal to the number of elementary k-flows. Those rerouting are costly, as the network need to stop working for a period of time during that operation [10]. When we have several thousands links in the network, we have to stop the network several thousands times during its lifetime. That can significantly reduce our network performance. Finding a linear combination that minimizes the number of times we have to stop the network is the motivation of this paper.

[1] The author would like to thank Mr. Jean-François Baffier, Mr. Wenkai Dai, Mr. Hidefumi Hiraishi, Mr. Bingkai Lin, Prof. Hiroshi Imai, and anonymous reviewers for giving several useful comments during the course of this research.

[2] Email: `vorapong@nii.ac.jp`

[3] In [4], Du and Kabadi claim that they can improve the computation time to $O(|E|^2)$. However, there are some typos in critical parts of their paper. Although we tried our best to correct those typos, we cannot find a way to fix them in this state.