CCIT Report #560

November 2005

Local Partition Hierarchies for General Graphs

Yitzhak Birk*, Idit Keidar*, Liran Liss*and Assaf Schuster†

December 1, 2005

Abstract

We present an algorithm for constructing a hierarchy of partitions with certain locality properties for general unweighted connected graphs. Every level in the hierarchy is a collection of disjoint sets of neighboring nodes, which we refer to as weak clusters. (Weak clusters are not necessarily connected.) The construction has two salient properties. First, it forms a refinement hierarchy, i.e., each (weak) cluster of a certain level is fully subsumed in some cluster of the next level. Second, every level is associated with a radius r and a slack function $\alpha(r) < r$, which characterize the sizes of its clusters. More specifically, r dictates an upper bound on a cluster's radius and $\alpha(r)$ is a lower bound on the minimal radius of some neighborhood that a cluster must cover. This construction servers as a building block for an efficient distributed aggregation algorithm. It may also suite other locality-sensitive algorithms based on hierarchal clustering, in which minimal cluster sizes are a concern.

1 Introduction

Graph constructions are an important building block for efficient locality-sensitive algorithms, which are essential for contemporary large-scale distributed systems. Such constructions are often called Locality-Preserving (LP) representations, as their structure "faithfully captures the topology of the network itself" [9]. Examples for LP representations and their role in distributed computing include using graph coloring for resource allocation [8], hierarchal covers for routing [1], and geographic partitions for resource allocation [6].

In this paper, we provide an LP representation for solving distributed aggregation problems. Specifically, we present a sequential algorithm, HPART, for constructing a hierarchal partition with special locality properties, which is required by the efficient I-LEAG aggregation algorithm [2]. Our construction is applicable to any connected graph G. Each level of the hierarchy provides three data structures: clusters of nodes that partition the graph, cluster representatives called pivots, and routing trees that span clusters.

At the lowest level, every node is its own cluster. Apart from the highest level, which has a single cluster that comprises the entire graph, every cluster is completely subsumed in some cluster of the next level. In level i, cluster sizes are determined according to a radius $r = \theta^i$, where θ is an algorithmic parameter. Every cluster is subsumed by a neighborhood of its pivot p with radius r, and subsumes a neighborhood (of p) with radius $\alpha(r)$, where $\alpha(r) < r$ is a slack function. Therefore, while clusters cannot grow too much in diameter, they are also guaranteed to cover a minimum area of the graph. We do not require clusters be connected, since the connectivity of G allows their nodes to communicate. Finally, every cluster is provided with a directed tree that connects its pivot with those of its subsumed clusters in the previous level. Based on these trees, a pivot can communicate with all nodes covered by its cluster.

*Electrical Engineering Department, The Technion – Israel Institute of Technology.
Email: {birk,idish}@ee,liranl@tx.technion.ac.il.

†Computer Science Department, The Technion – Israel Institute of Technology.
Email: {assaf}@cs.technion.ac.il.