COMBINATORICA
Akadémiai Kiadó – Springer-Verlag

COMBINATORICA 13 (4) (1993) 441–454

# LOW DIAMETER GRAPH DECOMPOSITIONS*

NATHAN LINIAL** and MICHAEL SAKS†

*Received October 6, 1990*

A *decomposition* of a graph $G=(V,E)$ is a partition of the vertex set into subsets (called *blocks*). The *diameter* of a decomposition is the least $d$ such that any two vertices belonging to the same connected component of a block are at distance $\leq d$. In this paper we prove (nearly best possible) statements of the form: Any $n$-vertex graph has a decomposition into a small number of blocks each having small diameter. Such decompositions provide a tool for efficiently decentralizing distributed computations. In [4] it was shown that every graph has a decomposition into at most $s(n)$ blocks of diameter at most $s(n)$ for $s(n)=n^{O(\sqrt{\log\log n/\log n})}$. Using a technique of Awerbuch [3] and Awerbuch and Peleg [5], we improve this result by showing that every graph has a decomposition of diameter $O(\log n)$ into $O(\log n)$ blocks. In addition, we give a randomized distributed algorithm that produces such a decomposition and runs in time $O(\log^2 n)$. The construction can be parameterized to provide decompositions that trade-off between the number of blocks and the diameter. We show that this trade-off is nearly best possible for two families of graphs: the first consists of skeletons of certain triangulations of a simplex and the second consists of grid graphs with added diagonals. The proofs in both cases rely on basic results in combinatorial topology, Sperner's lemma for the first class and Tucker's lemma for the second.

## 1. Introduction

In this paper, we investigate a problem in algorithmic graph theory that originated in the theory of distributed computing. The systems we are concerned with can be modeled as graphs whose nodes correspond to processors and whose links correspond to communication channels between certain processors. One of the basic difficulties in designing algorithms for such systems is determining the extent to which the actions of the processors must be coordinated, and accomplishing this coordination as efficiently as possible. The most naive approach is to centralize the network operation by appointing one of the processors as a coordinator for the whole network and having all processes act under the direction of the coordinator. Centralization has several advantages; it often simplifies the problem considerably and facilitates the development of distributed algorithms based on known serial algorithms. On the other hand, rigid centralization often degrades system performance because of delays in communication between the coordinator and the other

AMS subject classification codes (1991): 05 C 12, 05 C 15, 05 C 35, 05 C 70, 05 C 85, 68 Q 22, 68 R 10

* A preliminary version of this paper appeared as "Decomposing Graphs into Regions of Small Diameter" in Proc. 2nd ACM–SIAM Symposium on Discrete Algorithms (1991) 321–330.

** This work was supported in part by NSF grant DMS87-03541 and by a grant from the Israel Academy of Science.

† This work was supported in part by NSF grant DMS87-03541 and CCR89-11388.

processors of the system. This problem is particularly significant in networks with large diameter and non-negligible message transmission time.

For these reasons, considerable amounts of research in distributed algorithms has focused on finding ways to decentralize distributed computation. For many problems it is possible to design algorithms in which each node acts with knowledge only of the activity of nearby nodes, and these "local" activities combine together to produce a global solution to the problem. The extent to which this is possible is referred to informally as the *locality* of the problem. Exploiting locality for specific problems leads to algorithms that are among the most novel and interesting in the area, for example, the beautiful symmetry breaking techniques of Cole and Vishkin ([8]) which have been used for graph coloring ([11]). Limitations on locality were addressed in [13].

This leads to the following general problem: find techniques for the design of distributed network algorithms that can be used to exploit locality. One class of techniques that has been proposed involves partitioning the network into regions of small diameter, coordinating action in each region through a local coordinator and combining the partial solutions together. This natural methodology has been considered by a number of authors; its earliest explicit statement known to us is in Awerbuch's [3] analysis of time and communication trade-offs required to achieve network synchronization. In [4] it is further used to improve the time complexity of distributed deterministic maximal independent set (MIS) algorithms, for graph coloring and distributed breadth first search. More recently, the approach has been applied to distributed routing [5] for a distributed all-pairs-shortest-distance algorithm and other similar problems [1].

The above discussion motivates the following graph definitions:

Let $G$ be a graph. A subset $W$ of vertices will be called a *block* of $G$. The *strong diameter* of a block $W$, $SD(W)$ is the maximum diameter of any connected component of the graph $G_W$ induced on $W$. The *weak diameter* $WD(W)$ is the maximum distance in $G$ between two vertices of $W$ that belong to the same connected component of $G_W$. (The difference between strong and weak diameter is that when computing weak diameter we are allowed to shortcut through vertices not in $W$ and thus $WD(W) \leq SD(W)$.) A partition $\Pi$ of the vertex set of a graph $G$ into $\lambda$ disjoint blocks is called a $\lambda$-*decomposition* of $G$. The *strong diameter* $SD(\Pi)$ (*weak diameter* $WD(\Pi)$) of $\Pi$ is the maximum strong diameter (weak diameter) of any of its blocks.

For a given graph we are interested in finding decompositions into a small number of (possibly disconnected) blocks each of small (strong or weak) diameter. This problem was introduced (with somewhat different terminology) in [4] as an attempt to solve a major outstanding problem in the theory of distributed algorithms: is there a deterministic algorithm for finding a maximal independent set in a distributed network that runs in polylog time? There are various partial results known for this tantalizing problem: a randomized distributed algorithm that runs in expected polylog time (Luby [14] and Alon *et al.* [2]), and a deterministic polylog time algorithm for bounded degree graphs (Goldberg *et al.* [11]). It was noted in [4] that given a $\lambda$-decomposition of the graph, an MIS can be constructed in a sequence of $\lambda$ rounds each taking time $O(D\,\mathrm{polylog}(n))$ where $D$ is the strong diameter, through the following iterative procedure: after $i$ iterations there will be