Applied Mathematics and Computation 266 (2015) 244–258

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Rooted level-disjoint partitions of Cartesian products[☆]

Petr Gregor[a,*], Riste Škrekovski[b,c,d], Vida Vukašinović[e]

[a] Department of Theoretical Computer Science and Mathematical Logic, Charles University, Malostranské nám. 25, 11800 Prague, Czech Republic
[b] Department of Mathematics, University of Ljubljana, Jadranska 19, 1000 Ljubljana, Slovenia
[c] Faculty of Information Studies, Ulica talcev 3, 8000 Novo mesto, Slovenia
[d] FAMNIT, University of Primorska, Glagoljaška 8, 6000 Koper, Slovenia
[e] Computer Systems Department, Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

ARTICLE INFO

Keywords:
Broadcasting
Level-disjoint partitions
Cartesian product
Hypercube

ABSTRACT

In interconnection networks one often needs to broadcast multiple messages in parallel from a single source so that the load at each node is minimal. With this motivation we study a new concept of rooted level-disjoint partitions of graphs. In particular, we develop a general construction of level-disjoint partitions for Cartesian products of graphs that is efficient both in the number of level partitions as in the maximal height. As an example, we show that the hypercube Q_n for every dimension $n = 3 \cdot 2^i$ or $n = 4 \cdot 2^i$ where $i \geq 0$ has n level-disjoint partitions with the same root and with maximal height $3n - 2$. Both the number of such partitions and the maximal height are optimal. Moreover, we conjecture that this holds for any $n \geq 3$.

1. Introduction

An interconnection network is often modeled as an undirected graph in which the vertices correspond to processors and the edges correspond to communication links between the processors. Graphs which represent topological structure of such networks are required to possess elegant properties such as small degree and diameter, high connectivity, recursive structure, symmetry, etc. Examples of graphs with such structures are hypercubes, butterfly graphs, 2-torus graphs, etc.

Within such networks very often the data transmission is a bottleneck for possible delays. So constructing efficient broadcasting schemes is one of the critical points for high computing performances. Broadcasting is an information dissemination process that involves one node (often called a *source*) in a network sending pieces of information to all other nodes in the network. More about broadcasting, gossiping and other related tasks see e.g. [3,4,5–7]. An interested reader may find some results about these collective communication operations on hypercubes in [1,2,14].

Here we consider a communication problem that is distinct from the standard telephone model in the following rules:

($a1$) data needs to reach every node (broadcasting);
($a2$) at a time step, a node can receive data only from one neighbor node;
($a3$) at a call, an informed node may contact to any number of uninformed neighbor nodes.

☆ This research was supported by the Czech Science Foundation grant GA14-10799S, ARRS Program P1-0383, ARTEMIS-JU project “333020 ACCUS”, and by Creative Core FISNM-3330-13-500033.

* Corresponding author. Tel.: +420 221914140; fax: +420 221914323.
E-mail addresses: gregor@ktiml.mff.cuni.cz (P. Gregor), skrekovski@gmail.com (R. Škrekovski), vida.vukasinovic@ijs.si (V. Vukašinović).

http://dx.doi.org/10.1016/j.amc.2015.05.059
0096-3003/

Let us remark that without rule $(a2)$, a vertex may receive enormous amount of data all together from its neighbors which have to be processed due to $(a1)$ and that can cause a delay. Also note that $(a3)$ represents the Δ-port model [1] or the link-bound model [3].

A variation to the above model where in $(a3)$ it is required that a vertex can contact only one uninformed neighbor was recently a subject of extensive research. Under this assumption trajectory of data transmission induces Hamiltonian paths/cycles and each node appears in different positions in all these paths/cycles (so called mutually independent paths/cycles). In 2005, Sun et al. [16] proved that for any vertex s, the n-dimensional hypercube Q_n contains $n-1$ mutually independent s-starting Hamiltonian cycles if $n = 2, 3$; and n mutually independent s-starting Hamiltonian cycles if $n \geq 4$. They also proved that for any set of $n-1$ distinct pairs of adjacent vertices, Q_n contains $n-1$ mutually independent Hamiltonian paths with these pairs of vertices as endvertices. Mutually independent Hamiltonian paths and cycles in hypercubes with faulty edges were considered in [8,9,12,15,17]. Beside hypercubes this concept was also considered for wrapped butterfly graphs [11], 2-torus graphs with odd number of vertices [18], k-ary n-cubes with k being odd [10], and also in star-networks [13].

In what follows, we explain more formally our model. A *level partition* of a graph G is a partition $\mathcal{S} = (S_0, \ldots, S_h)$ of $V(G)$ into linearly ordered sets, called *levels*, such that $S_i \subseteq N(S_{i-1})$ for every $1 \leq i \leq h$. The number $h = h(\mathcal{S}) = |\mathcal{S}| - 1$ is called the *height* of $\mathcal{S}$. This concept naturally occurs in many situations. In this paper we use it to represent possible ways of spreading information across the graph. Starting at all vertices from the level S_0, at each step messages are sent from all vertices of the current level to all vertices in the next level through edges of the graph. Thus h is the number of steps needed to visit all vertices. We do not consider which particular edges are used as this is irrelevant in our scenario. Clearly, they can be arranged so that each vertex in level S_i is informed by a single vertex from level S_{i-1}.

In a particular case when the starting level S_0 is a singleton, say $S_0 = \{r\}$, we say that the level partition is *rooted* at r and the vertex r is called the *root* of $\mathcal{S}$. Rooted level partitions can be obtained, for example, from rooted spanning trees by taking the set of vertices at distance i from the root r in the spanning tree as the ith level of the partition. In fact, a rooted level partition corresponds to all the spanning trees with the same root and levels. Since the edge-sets of the spanning trees are of minor importance for us, we consider them to be equivalent and we identify rooted level partitions with rooted spanning trees. Thus, the data transmission induces spanning trees and each node appears in different heights in all these trees.

Furthermore, it is often desirable that multiple messages are sent simultaneously. At the same time we would like to avoid overloaded vertices. With this aim we introduce the following concept. Two level partitions $\mathcal{S} = (S_0, \ldots, S_{h(\mathcal{S})})$ and $\mathcal{T} = (T_0, \ldots, T_{h(\mathcal{T})})$ are said to be *level-disjoint* if $S_i \cap T_i = \emptyset$ for every $0 \leq i \leq \min(h(\mathcal{S}), h(\mathcal{T}))$. Level partitions $\mathcal{S}_1, \ldots, \mathcal{S}_m$ are said to be (mutually) level-disjoint if each two partitions are level-disjoint. Then we say that $\mathcal{S}_1, \ldots, \mathcal{S}_m$ are *level-disjoint partitions*, shortly *LDPs*. If every partition is rooted in the same vertex r and they are level-disjoint up to the starting level $\{r\}$, we say that $\mathcal{S}_1, \ldots, \mathcal{S}_m$ are *level-disjoint partitions with the same root r*.

LDPs allow simultaneous broadcasting of different messages (or different pieces of the same message) so that every vertex receives at most one message in each step. In other words, broadcast messages sent along level-disjoint partitions never meet at the same vertex at the same time if they are sent synchronously in parallel. The number of level partitions determines how many messages can be broadcasted simultaneously while the maximal height of such level partitions determines the overall time of the broadcasting. Hence a general aim is to construct for a given graph

- as many as possible (mutually) level-disjoint partitions; and
- with as small maximal height as possible.

In this paper we develop a general construction of LDPs for Cartesian products of graphs that is efficient both in the number of level partitions as in the maximal height. As an example, we show that the hypercube Q_n for every dimension $n = 3 \cdot 2^i$ or $n = 4 \cdot 2^i$ where $i \geq 0$ has n LDPs with the same root and with maximal height $3n - 2$, see Theorem 2. Note that there cannot be more than n LDPs with the same root in Q_n as the first level of each rooted level partition consists of neighbors of the root and Q_n has a regular degree n. Furthermore, it can be shown (see Proposition 15) that for n LDPs with the same root in Q_n at least one partition has height at least $3n - 2$. Thus both the number of LDPs and the maximal height are in this sense optimal. Moreover, we conjecture that the above result holds for any $n \geq 3$.

2. Preliminaries

Throughout the paper we restrict ourselves only to connected graphs. We use standard graph terminology and notation. For a set $S \subseteq V(G)$ let $N(S)$ denote the set of neighbors of S in the graph G. For an integer $n \geq 1$ the *hypercube* Q_n of dimension n is the graph with the vertex set $V(Q_n) = \{0, 1\}^n$ and edges between every two vertices that differ in precisely one coordinate. Clearly, Q_n is bipartite, n-regular, and vertex-transitive. We denote by 0^n the vertex $(0, 0, \ldots, 0)$ and for a vertex $r \in V(Q_n)$ we denote by $\bar{r}$ the *antipodal* vertex to r; that is, the unique vertex with $d(r, \bar{r}) = n$.

The *Cartesian product* $G \,\square\, H$ of graphs G and H is the graph with the vertex set $V(G) \times V(H)$ and edges between (u, v) and (u', v') whenever $u = u'$ and $vv' \in E(H)$, or $uu' \in E(G)$ and $v = v'$. Note that $d_{G \square H}((u, v), (u', v')) = d_G(u, u') + d_H(v, v')$ for any $(u, v), (u', v') \in V(G \,\square\, H)$. In particular, note that for any $n, m \geq 1$ the graph $Q_n \,\square\, Q_m$ corresponds to Q_{n+m}.

Our construction of level partitions of $G \,\square\, H$ assumes that we have level partitions $\mathcal{S}$ and $\mathcal{T}$ of G and H, respectively. The key idea is to alternate directions from G and H whereas in G we follow levels from $\mathcal{S}$ and in H we follow levels of $\mathcal{T}$. However, we must assure that all vertices are covered. For this purpose we need that in one of the level partitions $\mathcal{S}, \mathcal{T}$ we can follow levels from the current one not only to the next one but also to the previous one. We say that such level partition is bidirectional.