Combinatorics of k-ary n-cubes with Applications to Partitioning

Weizhen Mao[1] *and* David M. Nicol[2]

Dept. of Comp. Sci., College of William and Mary, Williamsburg, VA 23185, USA

ABSTRACT Many communication networks can be viewed as graphs called k-ary n-cubes, whose special cases include rings, hypercubes, and toruses. This paper explores combinatorial properties of such graphs—in particular, the characterization of the subgraph of a given number of nodes with maximum edge count. Applications of these properties to partitioning parallel computations will also be discussed.

1 Introduction

A *k-ary n-cube* is a graph that may be defined as follows. Each node is identified by an n-bit base-k address $b_{n-1} \dots b_i \dots b_0$; for every *dimension* $i = 0, 1, \dots, n-1$ it shares an edge with nodes $b_{n-1} \dots b_i \pm 1 (\mathrm{mod}\ k) \dots b_0$.

We can also define a k-ary n-cube recursively. First, we define a *ring* of k nodes labeled $0, 1, \dots, k-1$ to be a graph with edges between nodes i and $i+1(\bmod\ k)$ for $i = 0, 1, \dots, k-1$. When $k = 1$, a ring is a point. When $k = 2$, a ring is two nodes sharing an edge. When $k \geq 3$, a ring is a conventional ring. The recursive definition of a k-ary n-cube is as follows.

- A k-ary 1-cube is a ring of k nodes. Without loss of generality, we place the k nodes on a line, and call the leftmost node the 0^{th} position node and the rightmost node the $(k-1)^{st}$ position node.
- A k-ary n-cube contains k *composite subcubes*, each being a k-ary $(n-1)$-cube, placed from left to right. For every position $i = 0, \dots, k^{n-1} - 1$, edges between composite subcubes are defined by connecting all k i^{th} position nodes into a ring.

Further, a k-ary n-cube can also be viewed as an n-dimensional torus, which is a $\underbrace{k \times \cdots \times k}_{n}$ cube of grids with wrap-around edges.

The second and the third definitions of k-ary n-cubes provide two ways of drawing k-ary n-cubes. See Figure 1 for an example.

Table 1 shows special cases of k-ary n-cubes. We notice that the class of k-ary n-cubes contains many topologies important to parallel computations, including rings, hypercubes and toruses; hence a thorough study of k-ary n-cubes is worthwhile. The following combinatorial properties of k-ary n-cubes are easy to verify, except perhaps Property 1.5. We leave their proofs to the reader.

Property 1.1. *A k-ary n-cube has k^n nodes.*

Property 1.2. *A k-ary n-cube contains k composite subcubes, each being a k-ary $(n-1)$-cube, and the number of edges with endpoints in different composite subcubes is k^{n-1} for $k = 2$ and k^n for $k \geq 3$.*

[1]This work was supported in part by NSF grant CCR-9210372.

[2]This work was supported in part by NASA grant NAS1-19480 while the author was on sabbatical at the Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA 23681. It was also supported in part by NSF grant CCR-9201195.

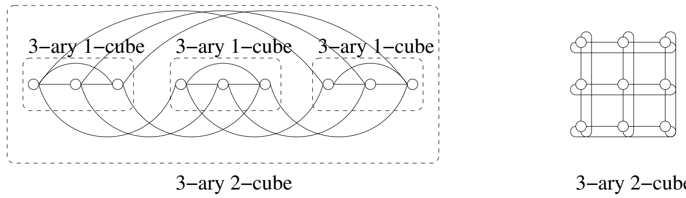

Figure 1: A 3-ary 2-cube

Table 1: Special cases of k-ary n-cubes

$k\backslash n$	1	2	≥ 3
1	point (ring)	point (torus)	point
2	edge (hypercube/ring)	square (hypercube/torus)	hypercube
≥ 3	ring	torus	k-ary n-cube

Property 1.3. *A k-ary n-cube is a regular graph, i.e., each node has the same degree. The degree of each node, $d_{k,n}$, is n for $k = 2$ and $2n$ for $k \geq 3$.*

Property 1.4. *The number of edges in a k-ary n-cube is nk^{n-1} for $k = 2$ and nk^n for $k \geq 3$.*

Property 1.5. *In a k-ary n-cube, for each i^{th} composite subcube $(0 \leq i \leq k-1)$ choose m_i nodes, and define $m = \sum_{i=0}^{k-1} m_i$. The number of edges with endpoints among these m nodes but in different composite subcubes is no larger than $t_k(m_0, \ldots, m_{k-1})$, where $t_2(m_0, m_1) = \min\{m_0, m_1\}$, and $t_k(m_0, \ldots, m_{k-1}) = \sum_{i=0}^{k-1} m_i - \max_{0 \leq i \leq k-1}\{m_i\} + \min_{0 \leq i \leq k-1}\{m_i\}$ for $k \geq 3$.*

Another important property of k-ary n-cubes that we shall devote the remainder of this paper to is about the relation between the number of nodes and the number of edges in any subgraph of a k-ary n-cube. A similar property related to VLSI concerns has been explored by Dally [2]. It is the *bisection-width* of k-ary n-cubes, the minimum number of edges one must cut when partitioning the graph into two subsets with equal numbers of nodes. In this paper we will consider a generalization of this notion: given that one of the subsets has exactly m nodes, what is the minimum number of edges between the subset and the rest of the graph?

We have previously studied properties of k-ary n-cubes in the context of load balancing [4]. Here graph nodes typically represent computation and edges represent communication. For any subgraph, define an *internal edge* to be one with two endpoints in the subgraph and an *external edge* to be one with one endpoint in the subgraph; viewing the subgraph as the set of nodes assigned to a processor, the number of external edges is a measure of the communication cost. Allowing nodes and edges to be weighted (reflecting relative computation and communication volumes, respectively), the "load" of a subgraph is taken to be the sum of the weights of its nodes and its external edges. If a k-ary n-cube is partitioned into P subgraphs, the *bottleneck* cost of the partition is the maximum load among all subgraphs. In [4] we showed that certain equi-partitions are optimal in the sense of minimizing the bottleneck cost, but that, surprisingly, there exist cases where the optimal partition is *not* an equi-partition. These results are based on a lower bound on a processor's communication cost, a bound that is achieved for selected subgraph sizes. The current paper completes that work by identifying an achievable bound for arbitrary subgraph sizes.

The problem of minimizing the external edge count (and hence the communication cost) of a subgraph with m nodes is the same as maximizing the number of edges contained in the subgraph (since k-ary n-cubes are regular graphs). Consider any subgraph S_m of $m \leq k^n$ nodes in a k-ary n-cube. Let $e(S_m)$ be the number of internal edges in S_m. Define the maximum number of internal edges in any subgraph S_m in a k-ary n-cube to be

$$e^*_{k,n}(m) = \max_{\forall S_m}\{e(S_m)\}.$$

We will say that a subgraph of a k-ary n-cube with m nodes is *optimal* if it has $e^*_{k,n}(m)$ internal edges.

We organize the paper as follows. In Section 2, we give a recursive method of computing $e^*_{k,n}(m)$ for $k = 1, 2, 3, 4$ and $m \leq k^n$. In Section 3, we study the case when $k \geq 5$. In Section 4, we mention some applications of our results to partitioning parallel computations. We conclude in Section 5.

2 Results for $k = 1, 2, 3, 4$

In this section we report results for cases $k = 1, 2, 3, 4$. The case of $k = 1$ is trivial: $e^*_{1,n}(m) = 0$ for $m \leq 1^n = 1$. Now consider $k = 2, 3, 4$. Recursively define function $F_k(m)$ as follows. It turns out that $F_k(m)$ exactly captures the quantity of interest.

Definition 2.1. *Let* $F_k(0) = F_k(1) = 0$ *and* $F_k(m) = (m \bmod k)F_k(\lceil \frac{m}{k} \rceil) + (k - m \bmod k)F(\lfloor \frac{m}{k} \rfloor) + t_k(\underbrace{\lceil \frac{m}{k} \rceil, \ldots, \lceil \frac{m}{k} \rceil}_{m \bmod k}, \underbrace{\lfloor \frac{m}{k} \rfloor, \ldots, \lfloor \frac{m}{k} \rfloor}_{k - m \bmod k})$ *for* $m \geq 2$.

Lemma 2.1. $F_k(m) \geq \sum_{i=0}^{k-1} F_k(m_i) + t_k(m_0, \ldots, m_{k-1})$ *for* $\sum_{i=0}^{k-1} m_i = m$ *and* $k = 2, 3, 4$.

Proof To save space, we will only prove the lemma for $k = 2$ since the proofs for $k = 3$ and $k = 4$ are too lengthy to be included in this paper. We induct on m. When m is small enough, the inequality certainly holds. For arbitrary m, consider the following three cases.

Case 1. Both m_0 and m_1 are even. Let $m_0 = 2a$ and $m_1 = 2b$ for some a and b. Therefore, $m = 2(a + b)$.

$$\begin{aligned} F_2(m) &= 2F_2(a+b) + (a+b) \quad \text{(Definition 2.1)} \\ &\geq 2(F_2(a) + F_2(b) + \min\{a, b\}) + (a+b) \quad \text{(Induction hypothesis)} \\ &= (2F_2(a) + a) + (2F_2(b) + b) + \min\{2a, 2b\} \\ &= F_2(m_0) + F(m_1) + t_2(m_0, m_1) \quad \text{(Definition 2.1)} \end{aligned}$$

Case 2. Both m_0 and m_1 are odd. The proof is similar to Case 1.

Case 3. One of m_0 and m_1 is even and the other is odd. Also similar to Case 1. □

Theorem 2.1. $e^*_{k,n}(m) = F_k(m)$ *for* $k = 2, 3, 4$.

Proof Since a k-ary n-cube contains k composite subcubes, each being a k-ary $(n-1)$-cube, assume that m_i nodes are chosen in the i^{th} composite subcube, for $i = 0, \ldots, k-1$ and $\sum_{i=0}^{k-1} m_i = m$. First we prove by induction on m that $e^*_{k,n}(m) \leq F_k(m)$. When

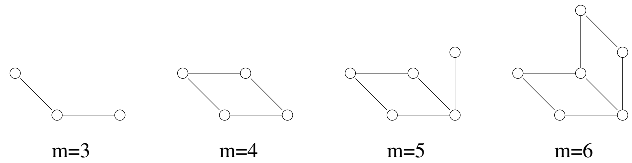

Figure 2: Subgraphs of a 2-ary n-cube achieving internal edge count $F_2(m)$

$m = 0, 1$, $e^*_{k,n}(m) = F_k(m) = 0$. Assume that the inequality holds for $\leq m-1$. Now consider m.

$$\begin{aligned} e^*_{k,n}(m) &\leq \max_{\forall \sum m_i = m} \{\sum_{i=0}^{k-1} e^*_{k,n}(m_i) + t_k(m_0, \ldots, m_{k-1})\} \quad \text{(Property 1.5)} \\ &\leq \max_{\forall \sum m_i = m} \{\sum_{i=0}^{k-1} F_k(m_i) + t_k(m_0, \ldots, m_{k-1})\} \quad \text{(Inductive hypothesis)} \\ &\leq F_k(m) \quad \text{(Lemma 2.1).} \end{aligned}$$

Next we show that there is a subgraph S_m with exactly $F_k(m)$ edges. Consider the k composite subcubes, each being a k-ary $(n-1)$-cube, in the k-ary n-cube. We allocate $\lceil \frac{m}{k} \rceil$ nodes into each of the first $m \bmod k$ composite subcubes and $\lfloor \frac{m}{k} \rfloor$ nodes into each of the remaining composite subcubes; the same method is then used recursively to allocate the nodes in each composite subcube. The total number of edges between these k subgraphs is given by Property 1.5. □

Figure 2 illustrates optimal subgraphs of a 2-ary n-cube for $m = 3, 4, 5, 6$.

3 Results for $k \geq 5$

Given that essentially the same approach defines the structure of optimal subgraphs for three successive values of k, one might suspect a general pattern for all k. It turns out that this is *not* the case. Consider the case of $k = 5$, $m = 6$. If we partition (in a certain dimension) into one subgraph of two nodes and four subgraphs of one node each we achieve six internal edges (a ring of five nodes, with one extra node hanging off the ring). However, it is possible to embed the six node graph illustrated in Figure 2 into the 5-ary n-cube, and achieve seven internal edges. An ability to embed subgraphs of a 2-ary n-cube into a k-ary n-cube turns out to be what is needed to characterize the optimal subgraphs of a k-ary n-cube with m nodes, when $k \geq 5$ and $m \leq 2^n$ (This second condition is needed in order to embed an optimal subgraph of a hypercube into k-ary n-cubes).

Lemma 3.1. *$F_2(m) \geq \sum_{i=0}^{k-1} F_2(m_i) + t_k(m_0, \ldots, m_{k-1})$ for $\sum_{i=0}^{k-1} m_i = m$ and any $k \geq 5$.*

Proof Similar to the proof of Lemma 2.1. □

Theorem 3.1. *$e^*_{k,n}(m) = F_2(m)$ for $k \geq 5$ and $m \leq 2^n$.*

Proof Similar to the proof of Theorem 2.1, using Lemma 3.1 instead of Lemma 2.1. □

What is $e^*_{k,n}(m)$ when $m > 2^n$? For this case we assume that either k is so large relative to m that an optimal subgraph cannot include wrap-around edges, or that the graph of

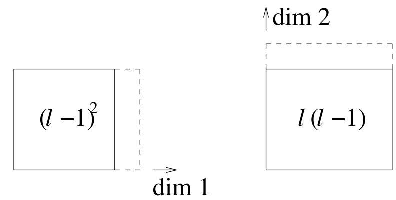

Figure 3: Construction of $\mathcal{C}_2(m)$

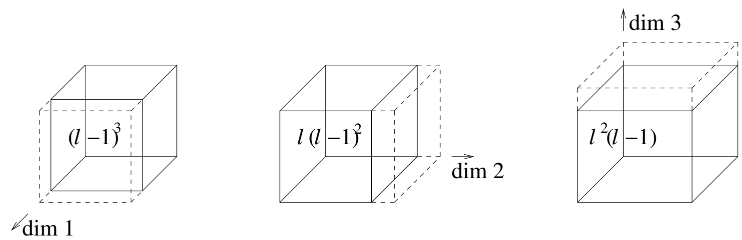

Figure 4: Construction of $\mathcal{C}_3(m)$

interest is a mesh (without wrap-around edges) whose local structure is like that of a k-ary n-cube. In other words, we now also consider multi-dimensional rectangular meshes, structures we will call n–D meshes.

Intuition tells us that the maximum number of internal edges $e^*_{k,n}(m)$ may be reached when the m nodes are placed as tightly as possible to form a "cubish" polyhedron. In this section, we report that this intuition is correct. In any dimension i, a subgraph of m nodes can be viewed as consisting of layers, each of which contains nodes with the same coordinate in dimension i. Furthermore, there may be edges between adjacent layers.

Definition 3.1. *Let m be such that $l^{i-1}(l-1)^{n-i+1} < m \leq l^i(l-1)^{n-i}$ for some $l \geq 2$ and some $1 \leq i \leq n$. Let $\delta = m - l^{i-1}(l-1)^{n-i+1}$. The n–D cubish polyhedron of m nodes in a k-ary n-cube, denoted as $\mathcal{C}_n(m)$, can be defined recursively as follows.*

- *$\mathcal{C}_1(m)$ is a line of m nodes.*
- *To construct $\mathcal{C}_n(m)$, we start with an $\underbrace{l \times \cdots \times l}_{i-1} \times \underbrace{(l-1) \times \cdots \times (l-1)}_{n-i+1}$ n–D rectangle (or cube if $i = 1$). For the remaining δ nodes, we construct an $(n-1)$–D layer $\mathcal{C}_{n-1}(\delta)$ and add it on the top of the n–D rectangle (or cube) in dimension i.*

The above procedure for constructing $\mathcal{C}_n(m)$ is very much like making a ball of yarn. The idea is to fill in each side (dimension) with yarn (nodes), one side (dimension) at a time. Figure 3 shows the construction of $\mathcal{C}_2(m)$, and Figure 4 shows the construction of $\mathcal{C}_3(m)$. Let $e_n(m)$ be the number of edges in the cubish polyhedron $\mathcal{C}_n(m)$. We next prove that $e_n(m) = e^*_{k,n}(m)$.

Theorem 3.2. *The edge count $e_n(m)$ in a cubish polyhedron $\mathcal{C}_n(m)$ is the maximum among all subgraphs S_m of m nodes in a k-ary n-cube (or in a n–D mesh), when wrap-around edges can be discounted.*

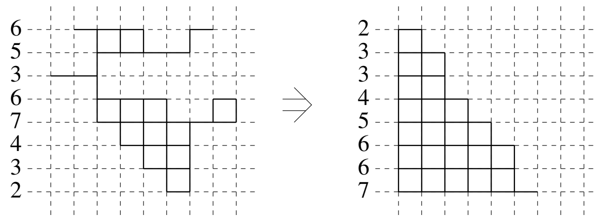

Figure 5: Rearrange S_m (in the k-ary 2-cube) without decreasing $e(S_m)$

Proof We prove by induction on n. When $n = 1$, the theorem is trivially true. Assume that the theorem holds true for $n - 1$. Now consider the case of n. Let m be such that $l^{i-1}(l-1)^{n-i+1} < m \leq l^i(l-1)^{n-i}$ for some $l \geq 2$ and some $1 \leq i \leq n$. Let $\delta = m - l^{i-1}(l-1)^{n-i+1}$. Let S_m be any subgraph of m nodes with $e(S_m)$ internal edges in the k-ary n-cube. We wish to prove that $e(S_m) \leq e_n(m)$.

We can view S_m as having several $(n-1)$-D layers of nodes stacked on each other in a certain dimension. Rearrange the order of the layers by sizes (node counts) and within each layer rearrange the nodes into an $(n-1)$-D cubish polyhedron. See Figure 5 for an example. The numbers in the figure are the sizes of the layers. If after this rearrangement there are h layers and s_i is the size of the i^{th} layer with $s_1 \leq s_2 \leq \cdots \leq s_h$, then by the inductive hypothesis we have

$$e(S_m) \leq (e_{n-1}(s_1) + s_1) + (e_{n-1}(s_2) + s_2) + \cdots + (e_{n-1}(s_{h-1}) + s_{h-1}) + e_{n-1}(s_h).$$

Note that $s_1 + s_2 + \cdots + s_{h-1}$ is the number of edges (legs) between adjacent layers.

We have a few observations about the new subgraph obtained. First, layers in each dimension (not just the dimension chosen in the rearrangement) are stacked on each other by sizes. Second, $h \geq l$. Assume that $h \leq l - 1$ for all dimensions. We must have $m \leq (l-1)^n$, which is impossible. Third, $s_1 \leq l^{i-1}(l-1)^{n-i}$. Suppose not. We must have $m = s_1 + \cdots + s_h \geq hs_1 \geq ls_1 > l^i(l-1)^{n-i}$, which is impossible.

Let us go back to the induction step, in which we assume that $e_{n-1}(m)$ is maximum and wish to prove that $e_n(m)$ is maximum. We need another induction on m to prove this. When $m = 1, 2$, $e_n(m)$ is obviously maximum. Assume that $e_n(j)$ is maximum for $j \leq m-1$. Now consider the case $j = m$. We know by the inductive hypothesis that

$$e(S_m) \leq (e_{n-1}(s_1) + s_1) + e_n(m - s_1).$$

By Definition 3.1, we know that $\mathcal{C}_n(m - \delta)$ is in fact an n-D mesh with $l^{i-1}(l-1)^{n-i+1}$ nodes. $\mathcal{C}_n(m - \delta)$ can also be viewed as having l (or $l - 1$ if $i = 1$) layers stacked on each other, where each layer is an $(n-1)$-D mesh and has L nodes. Clearly,

$$L = \begin{cases} (l-1)^{n-1} & \text{if } i = 1; \\ l^{i-2}(l-1)^{n-i+1} & \text{if } i \geq 2. \end{cases}$$

We can show that $s_1 < L + \delta$. Suppose not. We must have $m \geq hs_1 \geq ls_1 \geq lL + l\delta > lL + \delta \geq m$, which is impossible. To continue, we consider two cases.

Case 1. $s_1 \leq \delta$. We must have $l^{i-1}(l-1)^{n-i+1} \leq m - s_1 < l^i(l-1)^{n-i}$. Let $m - s_1 = l^{i-1}(l-1)^{n-i+1} + \delta'$. Then $s_1 + \delta' = \delta$. So

$$e_n(m - s_1) = (e_{n-1}(\delta') + \delta') + e_n(l^{i-1}(l-1)^{n-i+1})$$

and

$$e_{n-1}(s_1) + e_{n-1}(\delta') \leq e_{n-1}(\delta).$$

Therefore,

$$\begin{aligned} e(S_m) &\leq (e_{n-1}(s_1) + s_1) + e_n(m - s_1) \\ &= (e_{n-1}(s_1) + s_1) + (e_{n-1}(\delta') + \delta') + e_n(l^{i-1}(l-1)^{n-i+1}) \\ &\leq (e_{n-1}(\delta) + \delta) + e_n(l^{i-1}(l-1)^{n-i+1}) \\ &= e_n(m). \end{aligned}$$

Case 2. $s_1 > \delta$. This case is more complicated than the previous one. Due to the page limit of this paper, details are omitted but can be found in the full report [3]. □

4 Applications to Partitioning

There are different ways in which k-ary n-cubes are appropriate descriptions of parallel computations. One way is when at the lowest level the communication pattern of the computation is that of a k-ary n-cube. Another is when the communication patterns reflect a k-ary n-cube because the computation is *about* a k-ary n-cube. For instance, the computation may be a direct-execution simulation of an application running on an architecture whose communication network is a k-ary n-cube [6].

The results so far, especially the characterization of subgraphs with maximum internal edge count, despite having theoretical interest, have practical applications to partitioning. The first application arises when one wants to obtain an unequal bisection of a k-ary n-cube with the minimum number of edge cuts. Assume that one of the subgraph has m nodes. Then we have that the minimum number of edge cuts between the two parts of the k-ary n-cube is $md_{k,n} - 2e^*_{k,n}(m)$.

Our results may also be used in the context of computing lower bounding functions in branch-and-bound algorithms for partitioning. Consider a fine grained data parallel computation whose communication structure can be viewed as a k-ary n-cube, or related structure. The nodes of the graph are weighted individually to reflect computation costs, the edges of the graph are also weighted to reflect communication costs. We wish to find a *rectilinear* partitioning [5] of the graph into P subgraphs such that the bottleneck cost is minimized. A rectilinear partition is one in which the separating cuts are all hyperplanes of the form $x_i = c_{ij}$, a constant. Since finding the optimal rectilinear partition is intractable for dimensions larger than 2, branch-and-bound algorithms [1] are often used to obtain near-optimal partitions. The ability to compute $e^*_{k,n}(m)$ efficiently provides a way to establish a tight lower bounding function needed to direct the search in the branch-and-bound procedure. The details will not be discussed here.

Another application of our results is to identify optimal partitions (with respect to the bottleneck metric), even when those partitions are not entirely regular. Consider the problem of partitioning an 8-ary 2-cube (8×8 torus) into 13 subgraphs, assuming that all nodes have common computation weight w and all edges have unit communication cost. The problem clearly does not divide evenly. The minimal cost to a processor of having m nodes is $wm + C(m)$, where $C(m)$, the external edge count of an optimal subgraph with m nodes, is $4m - 2e^*_{8,2}(m)$; note that the cost function increases monotonically in m. The processor with the most nodes assigned will have at least $\lceil 64/13 \rceil = 5$ nodes. The optimal subgraph of the 8-ary 2-cube with 5 nodes is a square, with an attached singleton node. As illustrated in Figure 6, it is possible to nearly tessellate the 8-ary 2-cube with this optimal subgraph,

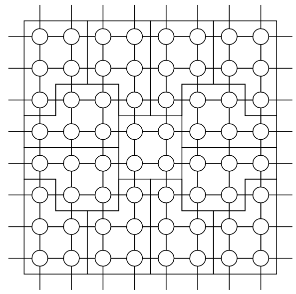

Figure 6: Optimal partition of an 8-ary 2-cube into 13 subgraphs

the only exception being one subgraph (the center square). The optimality of this partition derives from the fact that $wm + C(m)$ is monotone increasing in m, so that the bottleneck cost $\max\{wm_1 + C(m_1), \ldots, wm_{13} + C(m_{13})\}$ is minimized when the m_i's are nearly equal. The partition shown achieves the lower bound of $5w + C(5) = 5w + 10$.

5 Conclusions

This paper explores the combinatorial properties of k-ary n-cubes, and, in particular, describes how to construct subgraphs that are optimal in the sense of maximizing the number of internal edges, thus minimizing the number of external edges, given m nodes in the subgraph. While these results have combinatorial interest, they also have serious applications to partitioning parallel computations. k-ary n-cubes arise frequently in studies of parallel processing. The results and applications developed here help us to better understand these important graphs.

References

[1] G. Brassard and P. Bratley, *Algorithms: Theory and Practice*, Prentice-Hall, 1988.

[2] W. J. Dally, Performance analysis of k-ary n-cube interconnection networks, *IEEE Trans. on Comput.* **39**, 775–785 (1990).

[3] W. Mao and D. M. Nicol, On k-ary n-cubes: Theory and Applications, Tech. Rep., College of William and Mary, Williamsburg, Virginia, 1994.

[4] D. M. Nicol and W. Mao, On bottleneck partitioning of k-ary n-cubes, submitted for publication.

[5] D. M. Nicol, Rectilinear partitioning of irregular data parallel computations. *Journal of Parallel and Distributed Computing*, to appear.

[6] P. Dickens, P. Heidelberger and D. M. Nicol, A distributed memory LAPSE: Parallel simulation of message-passing programs. Proceedings of the 8^{th} Workshop on Parallel and Distributed Simulation, 32–38 (1994).